I BUS DI CAMPO ISOCRONI PROFIBUS DP-V2 E PROFINET I/O

Versione 0.9

Francesco Venturini

Indice

Capitolo 1
I BUS DI CAMPO

Introduzione

Nell'ambito industriale, e in particolar modo nei sistemi di automazione, sta diventando sempre più importante l'esigenza di poter trattare in tempo reale una crescente quantità di informazioni digitali.
Inoltre, l'aumento di unità "intelligenti" (quali PC, PLC nonché di sensori ed attuatori) all'interno dei cicli produttivi impone che vi sia un sistema di comunicazione aperto, cioè in grado di collegare tra loro più dispositivi prodotti da costruttori diversi, affidabile ed economico.
La risposta a queste problematiche è arrivata, da una decina d'anni a questa parte, con la comparsa nel mercato dei "bus di campo".
Questa soluzione tecnologica, prevede la presenza di un fascio di conduttori (il bus, appunto) al quale sono collegate tutte le apparecchiature. In genere la comunicazione tra i dispositivi è di tipo seriale e quindi un bus costituito da soli due fili risulta sufficiente allo scopo. La conseguenza immediata di questo tipo di struttura è la riduzione drastica dei cavi presenti aumentando quindi la facilità di cablaggio dei sistemi complessi, quali ad esempio, quelli di macchine utensili di grosse dimensioni. Oltre a questo l'uso di protocollo isocroni permette operazioni rapide, basate sul tempo per le quali la riproducibilità e il tempo reale (Real Time) sono le funzione più importanti. Attraverso i protocolli isocroni i loop di regolazione degli azionamenti possono essere chiusi anche tramite il bus di campo. L'isocronismo apre un gran numero di possibilità che non sono solo limitate ad applicazioni per azionamenti, viene infatti utilizzato anche per applicazioni con sensori e attuatori dove sia importante un trasferimento dati a intervalli di tempo di durata prefissata.

1.1 I BUS DI CAMPO

"Bus di campo" è un'espressione generica che descrive una nuova forma di comunicazione digitale dedicata ai sistemi a basso livello, quali sensori o attuatori, la quale si prevede sostituirà nell'industria la tecnologia analogica per collegamenti punto a punto basata su segnali a 4-20mA. La modalità di trasmissione numerica presenta, infatti, notevoli vantaggi in termini di possibilità di trasmettere più informazione e migliori potenzialità di interfacciamento. Già dai primi anni 80 i costruttori di sensori o attuatori di una certa complessità, come ad esempio telecamere o azionamenti, avevano fornito i propri sistemi di interfaccia numerica, seriale o parallela, per consentire una connessione punto a punto con i sistemi a più alto livello, quali PLC o PC. Tuttavia tale modalità di connessione, tipicamente una RS232 con un semplice protocollo ASCII, sebbene adatta per la fase di configurazione e predisposizione del trasduttore, durante la fase operativa nella quale il flusso informativo è più ridotto, comporta un numero di cavi confrontabile con la modalità analogica tradizionale. L'architettura a bus, nella quale un unico conduttore viene utilizzato per connettere più sistemi, consente invece una notevole riduzione dei cablaggi. Il bus rappresenta il mezzo fisico utilizzato per trasportare i dati. Tipicamente si tratta di un fascio comune di fili conduttori collegante assieme più dispositivi per permettere loro di comunicare scambiandosi dati. Contrariamente ad una connessione punto a punto, dove solo due circuiti si scambiano informazioni, un bus conta generalmente un numero di utenti superiore. Esistono molteplici modi per collegare insieme i vari elementi e, a seconda dell'applicazione, è consigliabile scegliere una topologia piuttosto che un'altra. Le più comuni sono quelle ad anello, a stella e a bus. Mentre una comunicazione punto a punto è limitata a due circuiti collegati ai due estremi del cavo, nel caso di un bus la flessibilità di estensione aumenta. In effetti, l'aggiunta di un nuovo trasduttore da collegare al bus esistente costituisce raramente un problema: la maggior parte dei bus di campo sono infatti in grado di rilevare autonomamente la lista degli utenti presenti. A livello di comunicazione, rispetto a connessioni tra soli due nodi, il modo di scambiare dati attraverso un bus di campo richiede regole più severe. Infatti, ad esempio, bisogna prevedere l'accesso contemporaneo al mezzo da parte di più elementi e

regolamentare nel modo più rigoroso possibile tutto ciò che serve per poter trasmettere i dati in maniera rapida ed affidabile. Tali regole di convivenza vengono comunemente chiamate protocollo. Un aspetto di rilevante importanza nel protocollo di tutti i bus di campo lo riveste la sicurezza dei dati trasmessi e ricevuti. Spesso sono previsti più meccanismi che agiscono in contemporanea e a volte, specialmente negli ambienti più critici, tale aspetto diventa un fattore discriminante nella scelta del protocollo da usare.

1.1.1 VANTAGGI INTRODOTTI DAL BUS DI CAMPO

I moderni sistemi di automazione industriale, in seguito all'integrazione di numerosi sensori e attuatori intelligenti, hanno la necessità di trasmettere oltre ai dati anche una serie di informazioni atte al controllo e supervisione della rete stessa. Tali informazioni devono essere disponibili in tempi brevissimi e in grado di viaggiare in entrambe le direzioni.
Risulta evidente che la tecnologia analogica tradizionale è insufficiente a tali scopi, tanto più che già con sistemi di medie dimensioni il problema del cablaggio rappresenta uno scoglio non indifferente, facendone lievitare i costi.
Brevemente i vantaggi introdotti dall'uso del bus di campo possono essere riassunti in:

- riduzione massiccia dei cavi e relativo costo;
- semplificazione della messa in servizio;
- possibilità di collegamento di prodotti di fabbricanti diversi;
- raccordo di moduli diversi su di una stessa linea;
- flessibilità di estensione;
- distanze coperte dal bus superiori a quelle raggiunte mediante cablaggio tradizionale;
- estensione dei campi di applicazione;

- riduzione dei costi globali;
- riduzione dei costi di engineering (una volta acquisita l'esperienza necessaria).

Comunque, esistono anche degli svantaggi che è corretto segnalare in contrapposizione ai vantaggi appena menzionati:

- necessità di conoscenze superiori;
- investimento in strumenti e accessori (sistemi di sviluppo, tool di monitoraggio e diagnosi, ecc.) abbastanza onerosi;

In realtà non si tratta di veri e propri svantaggi, in quanto essi rappresentano solo ciò che è necessario possedere per poter accedere a questa nuova tecnologia.

1.1.2 I SISTEMI DI COMUNICAZIONE IN AMBIENTE INDUSTRIALE

Solitamente troviamo almeno quattro livelli gerarchici in un sistema di comunicazione in ambito CIM (Computer Integrated Manufacturing, Fabbrica completamente automatizzata).

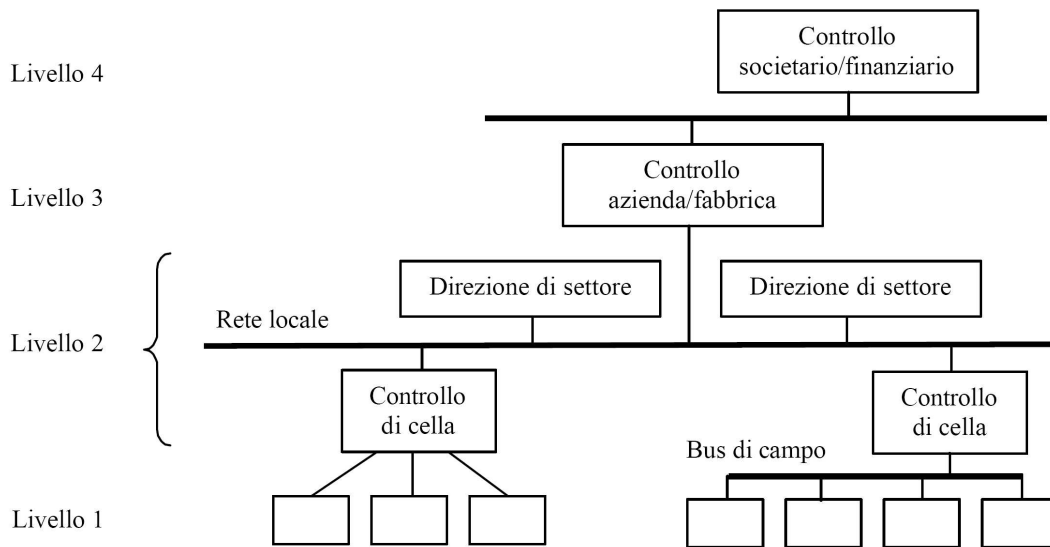



Figura 1.1– Livelli in un sistema di comunicazione in ambito CIM

Nel caso si analizzi l'organizzazione CIM del solo reparto di produzione di una azienda manifatturiera, è possibile individuare la tipica struttura, detta piramide CIM, visibile in figura seguente.

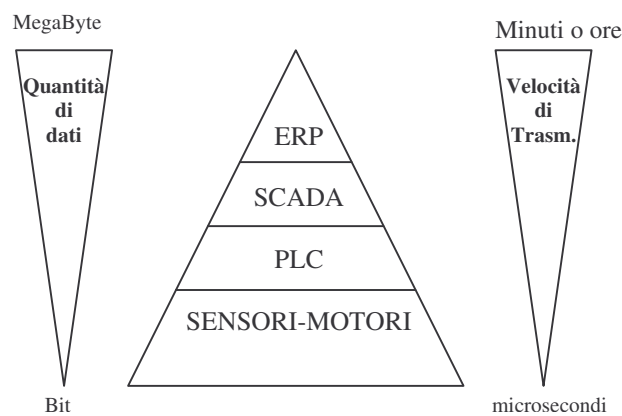



Figura 1.2 – La piramide CIM

Tipicamente in questo ambito il livello 1 è costituito dai trasduttori (finecorsa, sensori, encoder,…) e attuatori (relè, motori, elettrovalvole, spie di segnalazione,…) posti sull'impianto di produzione; a livello 2 ci sono i controllori di cella (PLC, CNC) che controllano sensori e trasduttori ai quali sono collegati direttamente o tramite bus di campo. Tutti i PLC che coordinano le varie celle di un reparto di produzione sono collegati fra loro tramite rete locale alla quale si connette anche un sistema SCADA (Supervisory Control And Data Acquisition, Sistema centralizzato di supervisione e controllo di sistemi distribuiti) di supervisione (livello 3). Il livello 3 viene anche detto livello di sistema e/o celle e i sistemi di comunicazione vengono detti bus di processo o di cella. A livello 4, detto anche livello di controllo e di servizio, ho in generale i sistemi CAD/CAM e, utilizzando una terminologia più moderna, i sistemi ERP/MES (Enterprise Resource Planning / Manufactoring Execution Systems). Tali sistemi elaborano i dati di produzione e li presentano in una forma adeguata al management, prelevandoli da tutti i sistemi SCADA della fabbrica ai quali sono connessi, sempre tramite LAN.
Salendo nella piramide aumenta la quantità di dati e diminuiscono i requisiti in termini di aderenza stretta al tempo reale.

1.1.3 I SISTEMI DISTRIBUITI

I sistemi distribuiti sono sistemi composti da un gran numero di CPU (PLC) distribuite in un impianto connesse da una rete ad alta velocità. Questi sistemi vengono di solito chiamati sistemi distribuiti, in contrapposizione ai sistemi centralizzati formati da una sola CPU, dalla sua memoria, dalle periferiche e da alcuni terminali.
Il fatto che sia possibile costruire sistemi distribuiti porta a vantaggi e a svantaggi; tra i vantaggi si possono enumerare: la distribuzione intrinseca di alcune applicazioni che richiedono macchine separate e distanti, affidabilità, scalabilità, modularità, la crescita incrementale, la condivisione dei dati, la comunicazione in tempo reale…; sebbene i sistemi distribuiti abbiano i loro punti di forza, essi hanno anche i loro

punti deboli, il principale dei quali è il software, molto più complesso che non quello dei sistemi centralizzati, e la dipendenza da un efficiente sistema di comunicazione in tempo reale.
Il vantaggio dell'automazione e della gestione computerizzata dei reparti è strettamente legato al concetto di integrazione e cioé una struttura aziendale viene snellita e resa più efficiente se i computer relativi ad una funzione possono comunicare in tempo reale con i computer relativi ad altre funzioni e magari dislocati in aree differenti. Ad esempio i computer addetti all'interfaccia uomo-macchina devono poter accedere a dati di tipo e provenienza diversa e tale "accesso" deve avvenire in tempo reale o perlomeno deve essere scandito da “tag” temporali; infatti per poter intervenire o anche solo analizzare un processo è necessario disporre dei dati $d_i=d_i(t)$ in modo da poterli correlare correttamente.

1.1.4 IL DETERMINISMO NEI BUS DI CAMPO

I primi bus di campo legati al mondo dei trasduttori sono stati sviluppati direttamente dai costruttori dei trasduttori stessi: si trattava in genere di reti master-slave basate ul livello fisico RS485. Lo standard EIA RS485, nato nel 1983 come raccomandazione tecnica dell’Electronic Industries Association (EIA), consente ad una stessa linea, Tipicamente costituita da un doppino, di ospitare fino a 32 ricevitori e fino a 32 trasmettitori con frequenze di trasmissione fino a 10 Mbaud e lunghezze del collegamento fino a 10 km. L’architettura master-slave, dove un solo master interroga ciclicamente gli slave, ossia i trasduttori, è la più semplice e quella che previene da situazioni di conflitto, ossia di accesso contemporaneo al bus da parte di più utenti. Ipotizzando comunicazioni bidirezionali, si possono avere due tipi di connessione:
1) utilizzando driver e receiver
2) utilizzando transceiver
Nel primo caso (figura 1.3 a sinistra), il master trasmette sulla linea Tx e riceve sulla linea Rx, mentre lo slave trasmette sulla linea Rx e riceve sulla linea Tx, proprio come avviene nei collegamenti RS232; nel secondo caso (figura a destra) vi è un’unica linea per cui viene a cadere la distinzione tra master e slave (half duplex)

tuttavia vi è un decremento delle prestazioni in quanto non vi può essere trasmissione e ricezione simultanea.

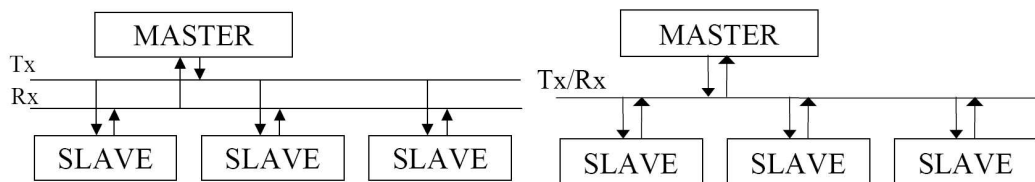



Figura 1.3 – Tipi di trasmissione full duplex – half duplex

La presenza di tanti protocolli diversi tra loro costringeva ad un notevole lavoro di implementazione sul sistema a più alto livello (PLC, PC), impedendo di fatto l'intercambiabilità tra componenti di differenti costruttori. Per evitare di dover ogni volta implementare i vari protocolli proprietari sul PLC, spesso si ricorreva a dei convertitori di protocollo, ossia dei sistemi che agivano come master della rete proprietaria dei trasduttori e come slave della rete nota al PLC.

Alcune di queste reti proprietarie master-slave sono evolute verso architetture multimaster, altre hanno modificato la strategia di accesso al bus per cui viene a cadere la distinzione tra utente master e utente slave. Nel primo caso (accesso al bus di tipo deterministico) i master sono autorizzati alla trasmissione per un tempo prestabilito (tempo di mantenimento del token), scambiandosi l'autorizzazione a trasmettere (token) secondo una sequenza prestabilita. Tale soluzione si dice **deterministica** perché è possibile calcolare un tempo massimo entro il quale l'informazione di qualsiasi utente verrà trasmessa: infatti il master avrà il token tipicamente ogni N*T (allocazione round robin del token), dove N è il numero dei master e T è il tempo di mantenimento del token per ogni master. Lo svantaggio di una simile configurazione è che manca una logica di priorità e cioè le segnalazioni urgenti possono essere inoltrate solo previo ottenimento del token. L'affidabilità di una simile rete è ottima, purché si preveda un meccanismo di sorveglianza del token e di eventuale reinizializzazione della rete. Nel secondo caso (accesso al bus di tipo casuale) ogni utente verifica se la rete è libera quindi accede alla rete e, in caso di collisione, tipicamente interrompe la trasmissione e attende nuovamente la

condizione di rete libera (retry). Tale strategia di accesso è non deterministica in quanto il tempo che intercorre tra l'esigenza di inoltrare un messaggio e l'effettiva presenza del messaggio in rete non è nota a priori ma dipende dal traffico sulla rete.

1.1.5 IL CONCETTO DI REAL-TIME

Come abbiamo visto, **determinismo** significa che un sistema reagisce *entro un intervallo di tempo prevedibile* (<100 msec).

Il **tempo reale**, nel contesto dei bus di campo, è legato al concetto di determinismo. Un sistema in tempo reale reagisce in un *arco di tempo definito più ristretto* (< 10 msec).

In una **rete isocrona** il trasferimento dati avviene in tempo reale ad intervalli di tempo equidistanti. Tale rete:

- garantisce il trasferimento di dati tra diverse stazioni entro un intervallo di tempo definito,
- consente un'esatta determinazione (previsione) del momento di trasferimento dei dati.

1.1.6 IL MODELLO ISO/OSI

L'architettura di una rete locale, come di una qualsiasi altra rete , può essere descritta in termini di architettura a strati secondo il modello OSI-RM (Open System Interconnection Reference Model), definito nel 1978 dalla ISO (International Standard Organization) e ispirato alla rete telefonica a lunga distanza. Tale modello stabilisce 7 livelli di descrizione:

Applicazione
Presentazione
Sessione
Trasporto
Rete
Collegamento dati
Fisico

Figura 1.4 – I livelli del modello ISO/OSI

Per una descrizione dettagliata del modello si rimanda all'appendice A.

1.1.7 PROFIBUS

PROFIBUS (**PRO**cess **FI**eld**BUS**) è un bus di campo fortemente voluto da *Siemens*, il cui sviluppo ed amministrazione è stato ufficialmente affidato ad un'organizzazione denominata *PROFIBUS Trade Organization* (*PTO*), composta da membri provenienti dal settore costruttivo, da istituti di ricerca e dall'utenza finale. Attualmente vi sono anche altre organizzazioni per lo sviluppo di PROFIBUS, quali *PROFIBUS User Organization* (*PNO*) e *PROFIBUS International* (*PI*), per l'Italia esiste il *PROFIBUS Network Italia* (*PNI*) con sede a Brescia.
La tecnologia PROFIBUS è nata come standard tedesco secondo la normativa DIN 19245 (1991), successivamente (1996) è stata riconosciuta conforme allo standard europeo (EN 50170). PROFIBUS è riconosciuto come protocollo di tipo 3 dalla normativa IEC (*International Electrotechnical Commision*) 61158 (*Digital data communication for measurement and control - Fieldbus for use in industrial control Systems*), mentre il protocollo di tipo 10 è riservato a *PROFINET*, la comunicazione basata su *Ethernet*. La norma IAC 61158 descrive i bus di campo secondo i livelli ISO/OSI, mentre la norma IEC 61784 (*Profile sets for continuous and discrete manufactoring relative to fieldbus use in industrial control systems*) descrive i profili che risiedono sopra il livello 7; PROFIBUS occupa il CPF3 (*Communication Profile Family*) e in particolare CPF3/1 si riferisce alla versione PROFIBUS DP, CPF3/2 si riferisce a PROFIBUS-PA mentre CPF3/3 è dedicato a *PROFINET*.

1.1.7.1 PROTOCOLLI DELLA FAMIGLIA PROFIBUS

La famiglia PROFIBUS consiste di tre protocolli compatibili tra loro:

- **PROFIBUS FMS** (*Fieldbus Message Specification*): la prima versione di PROFIBUS (1991), oggi in disuso e non previsto dalle norme IEC, rappresenta la soluzione "*general-purpose*" per la comunicazione anche a livello di cella. Il livello Applicazione è costituito dal *Fieldbus Message*
Specification (*FMS*) e dal *Lower Layer Interface* (*LLI*). FMS contiene il protocollo verso l'applicazione e le fornisce una vasta gamma di servizi. LLI comunica sopra

con FMS e sotto con lo strato di collegamento dati (livello 2), implementa relazioni di comunicazione che permettono a FMS di accedere al livello 2 indipendentemente dal tipo di dispositivo. Il livello 2 implementa il controllo per l'acceso al bus e garantisce la sicurezza dei dati. Per la trasmissione fisica, il protocollo FMS permette di utilizzare sia la tecnologia basata su RS 485 (codifica del bit tramite segnale su tensione differenziale) che le fibre ottiche.

- PROFIBUS DP (**Device Peripheral**): nato nel 1994 (DIN 19245 parte 3) è ottimizzato per collegamenti economici ad alte prestazioni. Questa versione PROFIBUS, che è la più utilizzata, è rivolta soprattutto alla comunicazione tra sistemi di controllo e dispositivi distribuiti di I/O. PROFIBUS DP utilizza gli strati 1 (PHY) e 2 (FDL) del modello a 7 strati ISO/OSI. Gli strati dal 3 al 7 non sono definiti per motivi di efficienza e le funzioni di comunicazione, previste dal protocollo e messe a disposizione dell'utente attraverso l'interfaccia utente, vengono mappate sul livello 2 da un applicativo denominato *DDLM* (*Direct Data Link Mapper*). Quest'architettura assicura una trasmissione veloce ed efficiente. Per la trasmissione fisica, vale lo stesso discorso fatto con FMS. DP e FMS, utilizzando la stessa tecnologia trasmissiva e lo stesso protocollo d'accesso al bus, possono operare simultaneamente sullo stesso cavo, anche se alle prestazioni del protocollo più lento (FMS).

- **PROFIBUS PA** (Process Automation): nato nel 1995 (DIN 19245 parte 4) e progettato principalmente per l'automazione di processo, consente di collegare sensori ed attuatori su una linea di comunicazione comune in aree a sicurezza intrinseca. Con il protocollo PA è possibile trasmettere dati e alimentazione su un bus a due conduttori in accordo con lo standard internazionale IEC 61158-2 (codifica del bit tramite segnale di corrente), che consente di operare in condizioni di sicurezza intrinseca permettendo ai dispositivi di essere alimentati direttamente sul bus. A livello fisico PROFIBUS PA può essere "interfacciato" con DP per mezzo di un bridge-accoppiatore. PA utilizza per la trasmissione dei dati un protocollo DP esteso oltre ad un Profilo PA nel quale viene definito il comportamento dei dispositivi di campo.

1.1.7.2 DISPOSITIVI IN RETI PROFIBUS

Per quanto riguarda i dispositivi, PROFIBUS distingue tra:

• Dispositivi ***MASTER***, che controllano la comunicazione sul bus. Un master può spedire messaggi senza richiesta esterna quando detiene il controllo della linea di comunicazione (possesso del *token*). Vengono anche chiamati stazioni attive.

• Dispositivi ***SLAVE***: sono unità periferiche che non possono accedere al bus direttamente, se non per divulgare informazioni diagnostiche che lo riguardano. Possono solamente riconoscere messaggi ricevuti o spedire messaggi al master se richiesti esplicitamente. Dal momento che richiedono piccole porzioni di protocollo, la loro implementazione è particolarmente economica. Vengono anche chiamate stazioni passive.

La gestione dell'accesso al bus viene implementata nel protocollo PROFIBUS secondo la filosofia master/slave, mentre per quanto riguarda l'accesso al bus tra master, si utilizza una procedura basata su ***token passing***. La procedura token passing garantisce che il permesso d'accesso al bus, concesso al master che possiede il token, sia definito per un intervallo di tempo preciso e costante; quindi il tempo che ogni master dovrà attendere per accedere al bus dipenderà dal numero di dispositivi attivi presenti nella rete e dal massimo tempo di utilizzo del bus permesso (*token hold time*). Il token viene passato da un master all'altro secondo un ordine prefissato (anello logico). La procedura master/slave consente alla stazione attiva che in un preciso momento possiede il token di accedere alle stazioni passive a lui assegnate. Il master può spedire messaggi agli slave o richiedere messaggi dagli slave.

1.1.7.3 FUNZIONI DEI LIVELLI DEL PROTOCOLLO PROFIBUS

Livello FDL

Durante la fase d'inizializzazione del sistema, il compito del sottostrato *MAC* (*Medium Access Control*, sottolivello del livello 2) di ogni dispositivo master è di definire il proprio indirizzo.
All'interno dell'anello logico costituito da tutti i dispositivi attivi. Durante la fase operativa, i master malfunzionanti o spenti devono essere tolti dall'anello logico, mentre quelli attivati successivamente devono essere inseriti nell'anello logico. Un altro compito importante del livello 2 (*FDL*) riguarda la sicurezza dei dati. Il formato delle trame utilizzato nel protocollo PROFIBUS assicura elevata integrità dei dati, grazie a telegrammi caratterizzati da distanza di Hamming HD=4 ottenuti secondo le norme dello standard IEC 870 5-1.

Livello DDLM

Lo strato di collegamento dati (*DDLM*) permette il trasferimento delle informazioni anche in modalità broadcasting, utilizzata per l'invio di trame a tutte le stazioni attestate sul bus, e multicasting per l'invio di dati ad un gruppo di stazioni.

1.1.7.4 TECNOLOGIE TRASMISSIVE

Il protocollo PROFIBUS, al fine di soddisfare diverse esigenze, in termini di velocità di trasmissione, distanza raggiungibile, sicurezza e possibilità di alimentazione lungo il bus, supporta diverse soluzioni tecnologiche: trasmissione *RS-485* per DP/FMS, trasmissione con fibre ottiche per DP/FMS, trasmissione IEC 1158-2 conforme, per PA.

RS-485

È la tecnologia trasmissiva più utilizzata nelle applicazioni Profibus e il suo impiego include tutti quei casi in cui si rende necessaria una trasmissione ad alta velocità

implementabile in maniera semplice. Utilizza come mezzo trasmissivo una coppia di conduttori attorcigliati e, se necessario, schermati. Si tratta quindi di una modalità di trasmissione molto economica e adatta a operare anche in ambienti sfavorevoli. La trasmissione del segnale avviene in corrente su tensione differenziale di 5V. Il transceiver di interfaccia viene protetto da eventuali situazioni di accesso simultaneo al segmento. Tutti i dispositivi sono connessi ad una struttura lineare e il medesimo segmento conduttore può ospitare fino a 32 unità. La modalità di connessione di Profibus è half-duplex e cioè ciascuna unità, master o slave, trasmette e riceve sulla stessa coppia di conduttori. Il bus è terminato all'inizio ed alla fine di ogni segmento da terminazioni attive. Quando si utilizzano più di 32 stazioni su più segmenti, si ricorre all'impiego di ripetitori (amplificatori di linea) per raccordare le varie sottoparti.

Il segnale è individuato da una tensione differenziale tra i due conduttori e la codifica è del tipo Non Return to Zero (NRZ). In particolare il simbolo '1' è rappresentato da una tensione differenziale positiva tra il pin 3 (RxD/TxD-P) ed il pin 8 (RxD/TxD-N) del connettore al bus, il simbolo '0' da un tensione differenziale negativa.

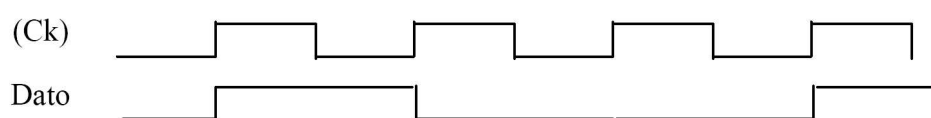



Figura 1.5

Trattandosi di trasmissione asincrona, il clock non viene trasmesso e i messaggi vengono inviati a byte, preceduti dal bit di start ('0') e seguiti da parità e stop ('1') (efficienza pari a 8/11).

Figura 1.6

FIBRA OTTICA

L'impiego di conduttori a fibra ottica può essere utile in ambienti ad alta interferenza elettromagnetica oppure per aumentare la distanza massima raggiungibile o la

velocità massima impiegabile. Il segnale digitale pilota un diodo emettitore di luce affacciato alla fibra ottica. All'altra estremità un dispositivo fotosensibile trasforma gli impulsi luminosi in impulsi elettrici. Ulteriori vantaggi in questo tipo di supporto riguardano la leggerezza, la resistenza meccanica, la larghezza di banda e quindi la velocità di trasmissione, che risulta inversamente proporzionale alla lunghezza del collegamento e non al suo quadrato, come nei conduttori metallici.
In particolare possono essere utilizzati due tipi di conduttori:

- Un cavo economico di accoppiamento in fibra ottica di plastica per interno in applicazioni di ridotte estensioni (distanze inferiori ai 50 m.).
- Un cavo LWL (dal tedesco Lichtwellenleiter che significa "conduttore di onde luminose") in fibra in vetro per interno ed esterno con distanze inferiori al chilometro.

Molti costruttori realizzano connettori speciali che integrano convertitori da fibra ottica a RS-485 e viceversa, e ciò semplifica l'impiego di questa tecnologia.

IEC 1158-2

Lo standard IEC 1158-2 risponde alle esigenze dell'industria chimica e petrolchimica che richiedono comunicazioni intrinsecamente sicure; si tratta di un protocollo basato sulla codifica dei bit per mezzo di segnali di corrente ed è spesso individuato con la sigla H1. La trasmissione è basata sui seguenti principi:

- ogni segmento ha una sola fonte di alimentazione, l'unità di alimentazione
- non si riversa potenza sul bus quando una stazione invia dati
- ogni dispositivo consuma una corrente di base (tipicamente 10 mA) nel suo stato di attesa
- ogni dispositivo si comporta come un 'pozzo' passivo di corrente
- ad entrambi i capi del bus sono presenti terminazioni passive (RC serie, R=100Ω, C=1μF)
- sono permesse reti lineari, ad albero o a stella
- è possibile aggiungere tratti ridondanti di bus per aumentare l'affidabilità.

I dispositivi devono essere alimentati con una corrente continua di almeno 10 mA, mentre il segnale trasmesso deve essere caratterizzato da una modulazione di 9 mA sulla componente continua. Il numero massimo di unità collegabili è 32, anche se il

numero in realtà è limitato dal tipo di protezione contro le esplosioni scelta. Va detto che la connessione di dispositivi alimentati sul bus e di dispositivi alimentati esternamente è possibile se questi ultimi sono dotati di un appropriato isolamento in accordo con lo standard EN50020. A livello fisico Profibus PA si interfaccia con Profibus DP per mezzo di bridge-accoppiatore.

1.1.7.5 PROFIBUS DP (Device Peripheral)

La versione **Device Peripheral** (**DP**) del protocollo PROFIBUS è stata pensata per la comunicazione tra sistemi di controllo dell'automazione (come PC e PLC) e dispositivi d'ingresso/uscita distribuiti (sensori, attuatori, ...) ed è ottimizzata per comunicazioni ad alte velocità e per connessioni poco costose. La maggior parte dello scambio di dati in questo contesto avviene in modo ciclico, tuttavia per permettere lo svolgimento di procedure di configurazione, di diagnostica o di gestione degli allarmi, il protocollo supporta anche funzioni di comunicazione aciclica. L'aumento significativo di velocità rispetto al protocollo FMS deriva sostanzialmente dall'utilizzo del servizio SRD (*Send and Receive Data*) del secondo livello del protocollo, che consente la trasmissione di dati di ingresso e di uscita in un singolo ciclo di messaggio.

PROFIBUS DP prevede due tipologie di master: *DPM1* per la gestione degli slave e *DPM2* per lo svolgimento delle funzioni di diagnostica e programmazione. Possono essere presenti più DPM1 ciascuno con il suo gruppo di slave anche se tra loro i DPM1 non comunicano. È possibile invece la comunicazione tra un DPM1 e un DPM2.

Comunque, la velocità elevata di trasmissione non rappresenta l'unico elemento che ha determinato il successo della versione DP del protocollo PROFIBUS. Procedure d'installazione e servizi semplici, potenzialità diagnostiche e trasmissioni senza errori sono caratteristiche altrettanto importanti per gli utilizzatori.

1.1.7.6 TIPI DI DISPOSITIVI

Ogni sistema PROFIBUS DP riconosce tre differenti tipi di dispositivi:
- **DP Master Class 1 (DPM1)**: è un controllore centrale che scambia informazioni con le stazioni decentralizzate a lui assegnate (DP slave) all'interno di un ben definito ciclo di messaggio. Gestisce la trasmissione dei dati utente e può comunicare con i dispositivi DP Master (classe 2). Tipicamente appartengono a questa classe PLC, PC o sistemi VME
- **DP Master Class 2 (DPM2)**: vengono utilizzati nella fase di configurazione del sistema DP o nelle necessarie operazioni di monitoraggio e diagnostiche. Appartengono a questa classe i dispositivi di programmazione, di configurazione e pannelli di controllo.
- **DP Slave**: sono stazioni periferiche (dispositivi di I/O, trasmettitori, *Human Machine Interface* (HMI), valvole, ecc...) che ricevono dati in input (richieste) e/o spediscono in output informazioni al controller (risposte). Sono indirizzabili da entrambi i tipi di master. I master sono spesso denominati come stazioni attive, mentre gli slave sono detti stazioni passive.

1.1.7.7 PROFIBUS DP-V0, DP-V1, DP-V2

Sono state rilasciate tre versioni del protocollo Versione 0, Versione 1, Versione 2.
Le caratteristiche di base nate come protocollo PROFIBUS DP sono state inserite nel protocollo rinominato PROFIBUS DP-V0 (PROFIBUS DP Versione 0).
Le caratteristiche di PROFIBUS DP-V0 sono:

- Trasmissione ciclica di dati utente tra DP master e DP slave
- Attivazione e disattivazione dinamica di singoli DP slave
- Verifica della configurazione dei DP slave
- Potenti funzioni diagnostiche (3 livelli gerarchici di messaggi diagnostici)
- Sincronizzazione degli ingressi e/o delle uscite (tramite i telegrammi SYNC e FREEZE).

Non è tuttavia possibile uno scambio di dati realmente isocrono, ciò è dovuto a ripetizioni di telegrammi causati da interferenze o dai meccanismi di diagnostica inseriti durante lo scambio dati ciclico.
Oltre a queste funzionalità di base (Profibus DP-V0), Profibus-DP prevede un'estensione (DP-V1) che permette anche la comunicazione aciclica.
La comunicazione aciclica e' normalmente una comunicazione non critica dal punto di vista del tempo. Questa comunicazione è utilizzata per parametrizzare e calibrare i dispositivi di campo e per lo scambio di informazioni diagnostiche. La versione 2 di Profibus (DP-V2) aggiunge funzionalità quali la possibilità di comunicazione tra slaves (DXB) utilizzando messaggi broadcast secondo una logica publisher-subscriber. Un'altra importante funzionalità introdotta da DP-V2 è il modo isocrono. Grazie ad un messaggio broadcast di "global control" tutte le stazioni possono sincronizzarsi con l'inizio del ciclo. In pratica il messaggio di global control delimita l'inizio del ciclo che prosegue poi con la fase di scambio dati. Grazie a DP-V2 è possibile far funzionare l'intero sistema secondo un orologio comune. Profibus DP-V2 verrà analizzato in dettaglio nel capitolo 3.

1.2 LA SINCRONIZZAZIONE E I BUS DI CAMPO

Da anni la decentralizzazione guadagna sempre più importanza nella realizzazione di moderni impianti di automazione.
Questo trend è motivato soprattutto dal prezzo, minor onere di cablaggio, installazione più facile. Oggi gli utenti richiedono soluzioni decentrate anche per il comando di macchine veloci. I processi di produzione e di lavorazione diventano sempre più veloci. Contemporaneamente crescono anche le esigenze relative alla precisione della produzione.
In quest'ottica vengono richiesti tempi di reazione di processo brevi, definiti e riproducibili: questo anche con la periferia decentrata. Questo significa che i segnali di periferia devono essere letti ed emessi in una griglia temporale equidistante ed essere sincronizzati con il programma applicativo.
In più il tempo intercorrente dal rilevamento di un segnale tramite la periferia decentrata fino alla corrispondente reazione sull'attuatore deve essere il più breve possibile e riproducibile con precisione.
Applicazioni che hanno queste esigenze sono per esempio

- Motion Control
- Sincronismo
- Regolazioni
- Programmatori a camme a base software
- Misura su diversi punti di misura
- Misura di numero di giri e portata

Si soddisfa quest'esigenza realizzando un accoppiamento diretto tra il ciclo equidistante, le unità di periferia ed il programma applicativo.
L'accoppiamento sincrono di una soluzione di automazione sul protocollo equidistante viene definita "**isocronismo**" e offre i seguenti vantaggi.
Operazioni rapide, per le quali la riproducibilità e il tempo reale (Real Time) è la funzione più importante, possono essere automatizzate anche con la periferia decentrata.

L'isocronismo apre un gran numero di possibilità che non sono limitate solo alle applicazioni per gli azionamenti. L'isocronismo è molto adatto per applicazioni in cui i sensori e gli attuatori sono distribuiti sulla macchina.

1.2.1 CICLI DI LAVORAZIONE NON SINCRONIZZATI (SENZA ISOCRONISMO)

Nelle strutture di automazione decentrate realizzate fino ad oggi molti cicli di lavorazione si svolgono in modo asincrono. Nella figura 1.11 è mostrato un ciclo di un processo non sincronizzato che usa come protocollo PROFIBUS DP

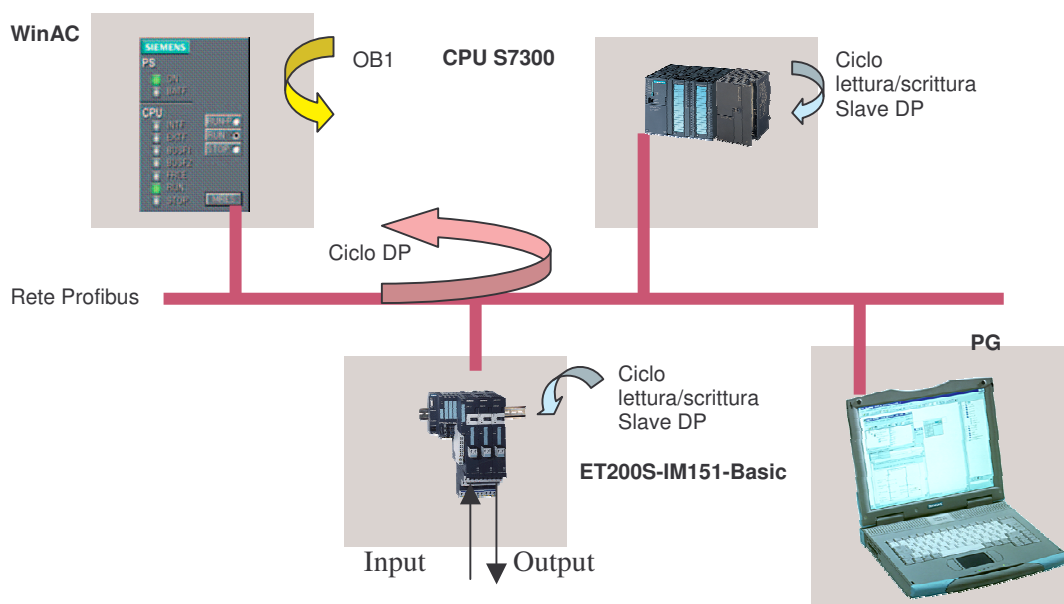



Figura 1.7 – Cicli non sincronizzati

Nell'applicazione in figura un master (WinAC) legge un ingresso da un modulo di periferia (ET200S), elabora il programma di controllo del processo e poi setta l'uscita dello slave.

Ciclo completo:

- lettura del segnale d'ingresso
- tempo di ciclo dell'ET 200S
- tempo di ciclo DP
- elaborazione del programma da parte della CPU
- tempo di ciclo dell'ET 200S

- emissione del segnale d'uscita

Vi sono almeno tre processi ciclici (ET 200S, ciclo PROFIBUS DP, CPU) non sincronizzati fra loro.
Come tempo di reazione si intende il tempo intercorrente tra il verificarsi di un evento e l'emissione di una reazione da parte di una unità di uscita. Questo tempo di reazione può essere pari alla somma dei singoli cicli e raggiungere anche il doppio.

1.2.2 CICLI DI LAVORAZIONE SINCRONIZZATI (ISOCRONISMO)

I protocolli isocroni garantiscono un trasferimento dati a intervalli di tempo di uguale lunghezza (equidistanti). E' importante come, oltre al protocollo, anche le applicazioni siano isocrone, cioè sincronizzate fra loro e a tempo di ciclo costante. Con riferimento all'esempio precedente di figura 1.11, si avrebbe tempo di reazione costante.
Isocronismo ha le seguenti proprietà:

- il programma applicativo è sincronizzato con l'elaborazione della periferia.
  Sincronismo significa che tutte le operazioni sono armonizzate tra loro nel tempo, tutti i dati in ingresso vengono prelevati in un istante definito.
  Anche i dati di uscita diventano attivi in un momento definito. I dati di ingresso e di uscita sono sincronizzati sul clock di sistema fino ai morsetti. I dati di un clock vengono sempre elaborati nel clock successivo.
- I dati di ingresso e di uscita vengono elaborati in modo equidistante. Equidistanza significa che i dati di ingresso vengono caricati in memoria a intervalli di tempo uguali ed i dati di uscita vengono emessi sempre a intervalli di tempo uguali.
- Tutti i dati di ingresso e di uscita vengono trasferiti in modo consistente.

Consistenza significa che tutti i dati dell'immagine di processo sono affini dal punto di vista logico e temporale.

1.2.3 APPLICAZIONE DELLE CARATTERISTICHE ISOCRONE

Come abbiamo visto un protocollo isocrono ha sia caratteristiche di sincronismo che di equidistanza. Analizziamo separatamente le applicazioni possibili sfruttando queste due caratteristiche.

Attraverso la caratteristica sincrona il rilevamento del valore istantaneo e l'emissione dei valori di setpoint avvengono in modo sincrono, cioè contemporaneamente per tutti gli ingressi o tutte le uscite, in modo da creare immagini di processo consistenti.

- Le applicazioni possono essere sincronizzate, poiché le singole posizioni vengono misurate contemporaneamente.
- I segnali con accoppiamento temporale molto stretto possono anche essere distribuiti nello spazio con la periferia decentrata, come ad esempio i segnali di avvio a più apparecchi per i quali la sequenza temporale è importante
- In conseguenza del rilevamento contemporaneo e del trasferimento sincrono, l'immagine della periferia risulta consistente.
  Questo consente il calcolo del rapporto di più valori analogici (per esempio più valori di pressione in una pressa)

Attraverso l'equidistanza il rilevamento del valore istantaneo e l'emissione dei valori di setpoint avvengono in modo equidistante, cioè sempre negli stessi intervalli di tempo.

- Si possono così effettuare calcoli basati sulla differenza di valore istantanei, per esempio nella misura di numeri di giri oppure misura di portata
- L'equidistanza può essere utilizzata nelle operazioni di dosaggio
- I loop di regolazione degli azionamenti possono essere chiusi anche tramite la periferia decentrata.

1.3 POSIZIONE DI MERCATO DI PROFIBUS

PROFIBUS è il bus di campo più diffuso a livello europeo e consente di collegare i tipici utenti di cella (*partner*) anche se di differenti case costruttrici: computer industriali, PLC, dispositivi di programmazione, controlli per robot e macchine utensili, sensori, attuatori, azionamenti, ecc.... Si tratta di un bus ad accesso deterministico. Il successo di PROFIBUS, che può vantare attualmente più di un milione di nodi installati ed una crescita vertiginosa in pochi anni, è dovuto al suo ampio spettro di applicabilità; dalla semplice connessione di attuatori e sensori fino al livello di cella, dall'industria manifatturiera al settore macchine utensili, dal settore del building al processo fino all'industria chimica e petrolchimica. È un bus di campo in grado di utilizzare, con lo stesso mezzo fisico, servizi per connessioni semplici orientate al byte e servizi per connessioni complesse (*frame* di configurazione, di parametrizzazione, di diagnostica ecc.) in modalità ciclica ed aciclica.

1.4 INTEGRAZIONE DI PROFIBUS CON PROFINET

Uno dei vantaggi di PROFINET sul mercato dipende, tra l'altro, dal fatto che le installazioni attuali di bus di campo possono essere espanse con PROFINET senza richiedere uno sforzo troppo grande.
Ciò significa che risulta possibile sviluppare un sistema che consiste di una miscela casuale di fieldbus e di sottosistemi basati su Ethernet.
E' così possibile un trasferimento continuo di tecnologia dai sistemi basati su fieldbus verso PROFINET.
In considerazione del gran numero di sistemi di PROFIBUS in uso, è assolutamente essenziale, nell'interesse di proteggere gli investimenti, permettere a questi sistemi di essere conglobati in PROFINET facilmente e senza necessità di essere adattati (fig.1.12).

Anche se PROFINET differisce concettualmente dai sistemi di bus di campo come PROFIBUS, esso è riuscito a definire delle strategie di transizione dalla rete Ethernet ad un bus di campo completamente trasparenti.
Le parti attuali del sistema non devono essere adattate.

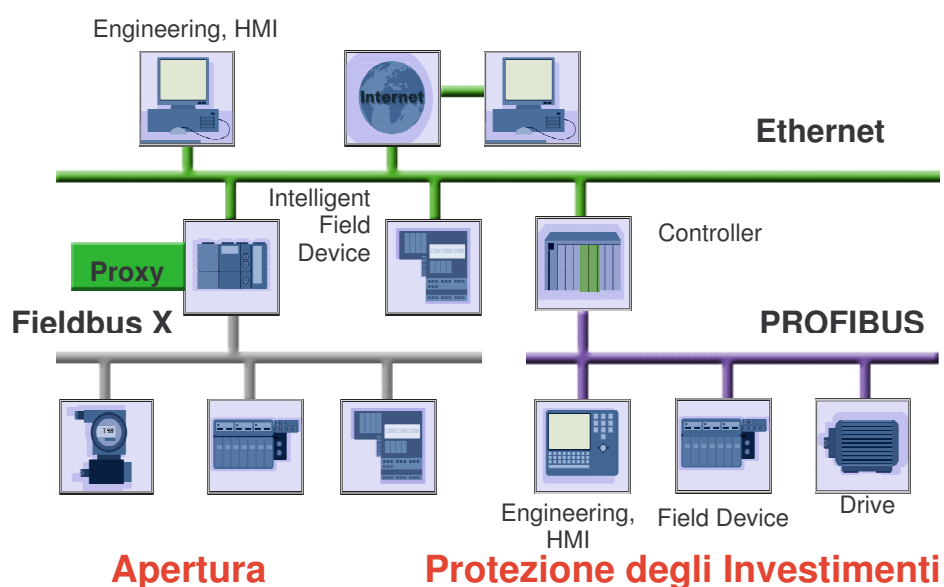



Figura 1.12 PROFINET offre apertura e protezione degli investimenti

La spinta verso una soluzione di integrazione semplice e trasparente proviene da più parti:

- L'utilizzatore finale gradirebbe un modo facile per integrare le sue installazioni attuali in un concetto di automazione PROFINET .
- Il costruttore dell' impianto vorrebbe usare la propria gamma di dispositivi senza la necessità di eseguire degli adattamenti su di essi per renderli compatibili con i progetti di automazione PROFINET.
- Il fornitore del dispositivo o un OEM vorrebbe potere integrare i suoi dispositivi attuali di campo senza sostenere spese per riadattarli ai sistemi PROFINET.

Capitolo 2
PROTOCOLLI ISOCRONI

2.1 BUS DI CAMPO E FUNZIONALITÀ DI SINCRONIZZAZIONE

Negli impianti automatizzati di ultima generazione, gli azionamenti elettrici, i convertitori AC, i servo controllori da quelli più semplici a quelli di alte prestazioni, vengono sempre più connessi a sistemi di controllo ad anello aperto o ad anello chiuso ed a sistemi di monitoraggio attraverso interfacce digitali collegate a un bus di campo.

Nei sistemi attuali la velocità dell'interfaccia di comunicazione è standardizzata. Al fine di utilizzare, nelle applicazioni di "motion control", bus di campo con "automazioni distribuite", gli standard dei bus di campo sono stati dotati di nuove funzionalità, tra le quali:

- Sincronismo del ciclo di clock. Se per gestire l'interpolazione e il controllo ad anello chiuso della posizione di un asse viene utilizzato un sistema di motion control centralizzato, allora l'anello di controllo deve essere chiuso attraverso il bus. Affinché il guadagno di anello sia abbastanza alto per soddisfare le prestazioni dinamiche, il ritardo deve essere minimo e specialmente costante. L'applicazione che controlla il movimento richiede il coordinamento di diversi assi, i valori di posizione reali dei diversi assi devono essere acquisiti in modo preciso nello stesso istante di tempo e valutati in sincronismo dal controllore; inoltre i valori di setpoint devono essere applicati a tutti gli assi nello stesso istante di tempo. I valori reali acquisiti, trasferiti e l'attivazione dei setpoint sono sincronizzati con il ciclo di clock insieme al controllo di posizione ad anello chiuso.

- Comunicazione da slave a slave. Rappresenta lo stato dell'arte nel campo dell'automazione, attraverso la comunicazione slave-to-slave gli azionamenti possono utilizzare il concetto di "automazione distribuita". Nel campo dell'automazione distribuita le applicazioni di controllo ad anello aperto o ad anello chiuso, che erano nel passato implementate da un'unità centrale, possono essere distribuite tra azionamenti intelligenti. Ad esempio un azionamento può essere asservito ad un altro ricevendone il riferimento di corrente mentre la regolazione di corrente avviene solo sull'azionamento principale. Se vogliamo che le funzioni di automazione siano decentralizzate e distribuite, i dati devono poter essere trasferiti direttamente tra gli azionamenti.
- Comunicazione aciclica. Per esempio, in PROFIBUS, i servizi aciclici sono utilizzati per trasferire richieste di parametri per il controllo delle operazioni e per il monitoraggio degli azionamenti in parallelo allo scambio di dati ciclico

Fino a quando le funzionalità dei bus di campo non vennero estese attraverso i protocolli isocroni non era possibile soddisfare tutto le richieste sopraelencate utilizzando un solo sistema. Negli impianti automatizzati che includevano azionamenti venivano spesso utilizzati in parallelo differenti sistemi con differenti bus di campo. Se, oltre al controllo dell'azionamento, esistevano nell'impianto periferiche con I/O distribuiti oppure postazioni per il controllo del processo, il tutto veniva implementato utilizzando bus differenti (in aggiunta al bus attraverso il quale veniva controllato l'azionamento, veniva utilizzato un sistema proprietario per la sincronizzazione degli azionamenti).
Le funzionalità dei bus di campo sono state estese per coprire tutte le richieste di funzionamento degli azionamenti elettrici utilizzando un sistema con un unico bus di campo.
Nel caso del protocollo PROFIBUS questo è stato reso possibile dall'estensione dello standard attraverso le specifiche PROFIBUS DP-V2.
Come ulteriore sviluppo nel campo dei protocolli per la sincronizzazione nel caso di reti Ethernet è stato proposto lo standard IEEE 1588-2002 ("IEEE Standard for a Precision Clock Synhronization Protocol for Networked Measurement and Control System")

L'introduzione di queste nuove funzionalità porta ad una riduzione dei costi di ingegnerizzazione, una ulteriore riduzione dei costi di servizio, manutenzione e del costo totale dei dispositivi utilizzati.

2.2 SINCRONIZZAZIONE DISTRIBUITA: IEEE1588

Lo standard IEEE1588 definisce un protocollo (PTP Precision Time Protocol) per la sincronizzazione di oggetti distribuiti su una rete. Questo protocollo può essere usato per sincronizzare i clock di sistemi di misura o di sistemi di controllo. Molti protocolli per real-time Ethernet supportano la IEEE1588 come metodo per la distribuzione di un riferimento di tempo unico per tutta la rete. In questo modo è possibile eseguire delle operazioni distribuite (campionamento, attuazione) svincolandosi dai ritardi introdotti dal mezzo trasmissivo. PTP può operare su almeno 24 tipi di bus inclusi Ethernet, Profibus, Devicenet etc. .
Secondo lo standard, gli oggetti di una rete sincronizzata tramite PTP possono essere:

- Ordinary clock: sono i componenti che possono comunicare con gli altri clock tramite un singolo percorso (fanno parte di una sola rete)
- Boundary clock: sono i componenti che possono comunicare con più reti diverse (sono l'equivalente dei gateway, ossia sono il punto di unione di due sottoreti diverse)

Ogni oggetto che dispone di un clock ha anche un grado di qualità del proprio clock, per esempio se il clock deriva da un orologio atomico o da un ricevitore GPS si ha la qualità massima. In base alla qualità del clock viene scelto nella rete il "Master clock", cioè il componente che distribuisce il proprio clock agli altri, chiamati "Slave clock".
La sincronizzazione si basa sullo scambio di 4 tipi di messaggi (Sync, Follow-Up, Delay-Request, Delay-Response)

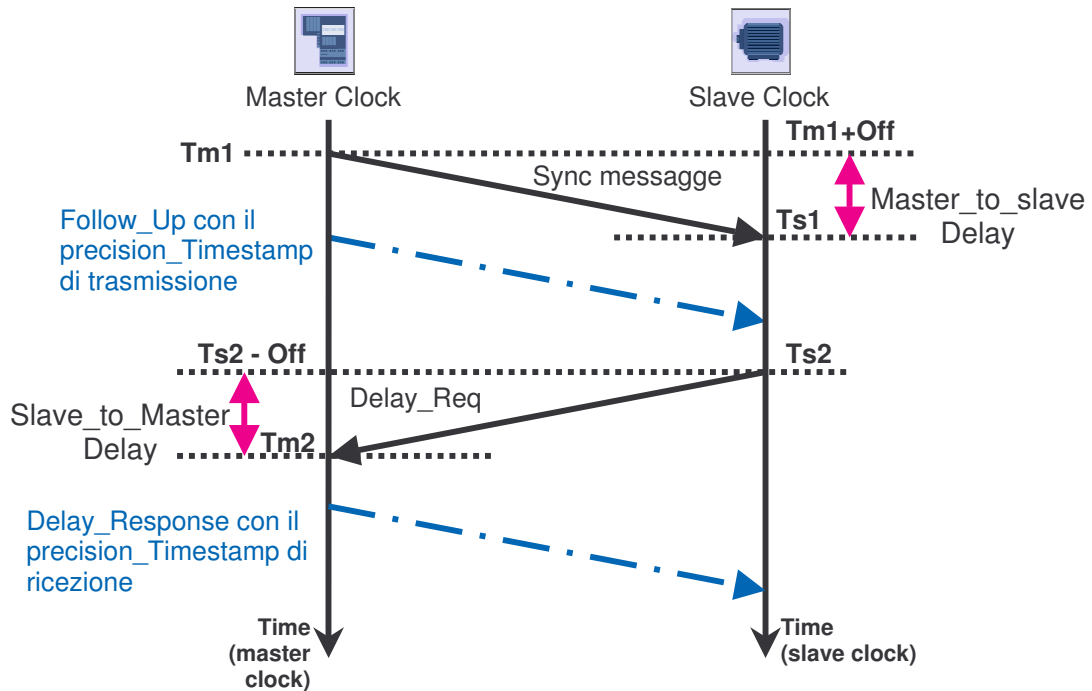



Figura 2.1 – Scambio dei messaggi per la sincronizzazione

Il master clock e lo slave clock possono essere sfasati di un certo offset chiamato Off, ossia :

Ts=Tm+Off

Quando il master clock invia il messaggio di Sync a uno slave rileva l'istante esatto di spedizione Tm1. Questo tempo (precision_timestamp di trasmissione) può essere messo nello stesso messaggio Sync oppure in un messaggio di tipo Follow-Up spedito successivamente. Quando lo slave clock riceve il Sync rileva l'istante di ricezione Ts1 e così può impostare la prima equazione:

Ts1= Tm1 + Off + master_to_slave_delay

In seguito lo slave clock spedisce un messaggio di Delay_request al tempo Ts2 che arriva al master nell'stante Tm2. L'istante di ricezione Tm2 viene anche comunicato allo slave per mezzo del messaggio Delay_response.

Si può quindi impostare un'altra equazione

$$Tm2 = Ts2 - Off + slave_to_master_delay$$

A questo punto lo slave calcola:

1. master_to_slave_delay_stimato = Ts1 – Tm1
2. slave_to_master_delay_stimato = Tm2 – Ts2
3. one_way_delay =
   = (master_to_slave_delay_stimato + slave_to_master_delay_stimato)/2 =
   = { (Ts1 – Tm1) +(Tm2 – Ts2) } / 2 =
   = {(Off + master_to_slave_delay) + (slave_to_master_delay-Off) } =
   = (master_to_slave_delay + slave_to_master_delay)/2
4. Off = Ts1 – Tm1 – one_way_delay

Naturalmente questi calcoli sono validi solo se il ritardo introdotto sulla linea di comunicazione da e verso il master è costante. Questo in una rete Ethernet non è sempre verificato poiché gli switch potrebbero introdurre dei ritardi asimmetrici a causa delle condizioni di traffico.

Per questo motivo i produttori di sistemi Ethernet real-time hanno proposto delle integrazioni allo standard. Per esempio PROFINET ha introdotto il concetto di "bypass clock" associato ai suoi ASIC ERTEC. Questi componenti si comportano come degli switch per quanto riguarda il traffico di rete normale, mentre per quanto riguarda il traffico PTP sono in grado di correggere al volo il contenuto dei messaggi di Sync. Infatti le correzioni dei timestamp saranno in grado di compensare le asimmetrie nei ritardi di comunicazione.

2.2.1 ISOCRONIA NEL CAMPO DEGLI AZIONAMENTI. L'AUTOMAZIONE DISTRIBUITA.

Il concetto di **automazione distribuita** prevede che le applicazioni di controllo ad anello aperto o ad anello chiuso, implementate nel passato da un'unità centrale, possono essere distribuite tra azionamenti intelligenti. Questo comporta un trasferimento dei dati direttamente tra gli azionamenti.

In PROFIBUS vengono definite sei classi di applicazioni per l'integrazione degli azionamenti in sistemi automatizzati:

- Azionamento Standard con funzioni di automazione nell'unita' di controllo centrale (Applicazione di Classe 1)
- Azionamento Standard con funzioni di automazione distribuite anche negli azionamenti (Applicazione di Classe 2)
- Azionamento distribuito per singoli assi e funzioni di automazione nell'unita' di controllo centrale. Ogni asse è controllato da un azionamento che controlla interpolazione della traiettoria, posizione e velocità. (Applicazione di Classe 3)
- Controllo di posizione e interpolazione della traiettoria centralizzato (Applicazione di Classe 4)
- Controllo di posizione distribuito con interpolazione della traiettoria centralizzata (Applicazione di Classe 5)
- Controllo del movimento per processi sincronizzati (Applicazione di Classe 6)

Per chiarire il concetto di automazione distribuita vengono di seguito riportate tre applicazioni, una che utilizza la comunicazione ciclica di PROFIBUS per comandare un azionamento standard e due che necessitano per il funzionamento l'utilizzo di un protocollo isocrono come PROFIBUS DP-V2.

2.2.1.1 AZIONAMENTO STANDARD (CLASSE 1)

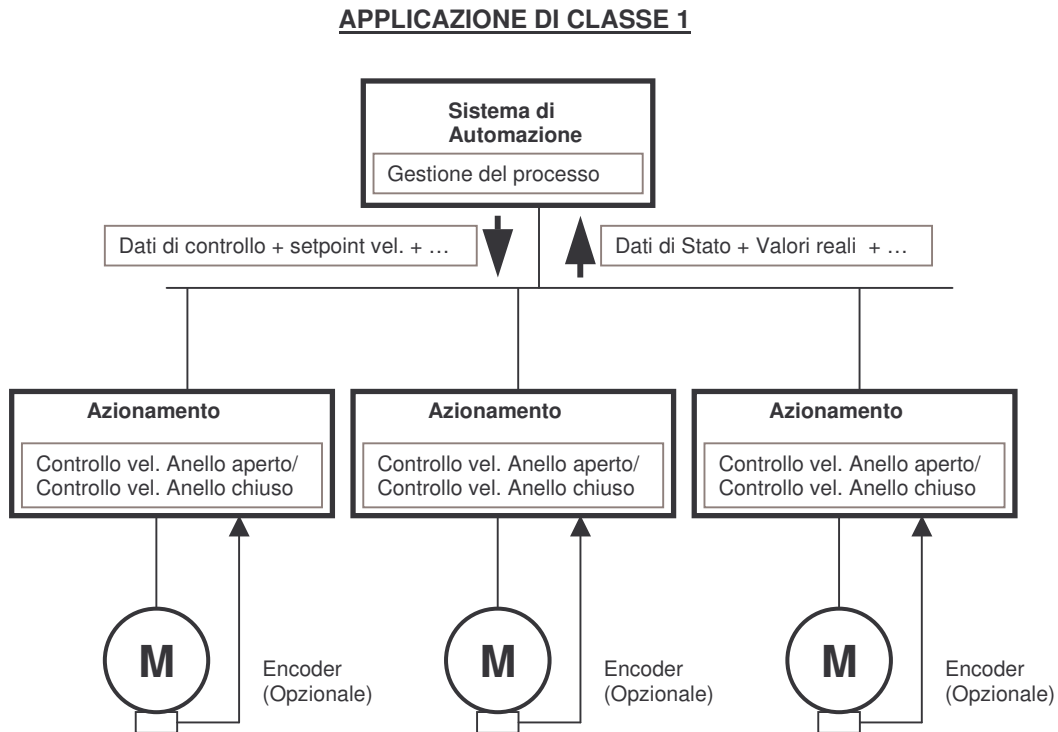



Figura 2.2 – Azionamento Standard

Nei casi più semplici, l'azionamento è controllato tramite l'invio di un setpoint attraverso il bus (per esempio inviando il setpoint di velocità). La regolazione della velocità è implementato direttamente nell'azionamento. Il PLC si preoccupa di tutte le funzioni per automatizzare il processo. Il bus di campo è semplicemente il mezzo trasmissivo tra il sistema di automazione e l'azionamento. In PROFIBUS DP si utilizza la comunicazione ciclica standard per lo scambio dati.
Questo tipo di approccio è usato principalmente per applicazioni poco critiche, con il PLC normalmente impiegato come sistema di automazione ed eventualmente di controllo di processo a più alto livello. L'utilizzo della sincronizzazione del clock e della comunicazione da slave a slave non è necessaria per questa classe di applicazioni.

2.2.1.2 CONTROLLO DI INTERPOLAZIONE DELLA TRAIETTORIA E DI POSIZIONE CENTRALIZZATO (CLASSE 4)

APPLICAZIONE DI CLASSE 4

Sistema di Automazione

Gestione del processo

Controllo della posizione

Dati di controllo + Setpoint vel. + …

CLOCK

Dati di Stato + Posizione Attuale + …

SINCRONISMO DI CLOCK

Azionamento

Controllo di velocità ad anello chiuso

Azionamento

Controllo di velocità ad anello chiuso

Azionamento

Controllo di velocità ad anello chiuso

M

Encoder

M

Encoder

M

Encoder

Figura 2.3 – Azionamento di classe 4

Gli azionamenti per manipolatori o per applicazioni nel campo della robotica spesso richiedono una sequenza di movimenti coordinati per i vari bracci. Il controllo del moto è implementato principalmente in una unità di automazione centralizzata. L'unità calcola per ogni azionamento speciali profili di setpoint. Attraverso il movimento coordinato di diversi azionamenti si possono implementare le traiettorie desiderate. In aggiunta alle normali funzioni per automatizzare il processo, l'unità di controllo centrale deve includere funzioni di posizionamento e di interpolazione della traiettoria. I valori dei setpoint e i valori reali sono trasferiti attraverso il bus in modo isocrono. Nell'azionamento è incluso solo l'algoritmo per il controllo ad anello chiuso della velocità. Dal momento che la posizione è controllata attraverso il bus di campo, la sincronizzazione del bus deve essere molto precisa.

2.2.1.3 CONTROLLO DEL MOVIMENTO PER PROCESSI SINCRONIZZATI (CLASSE 6)

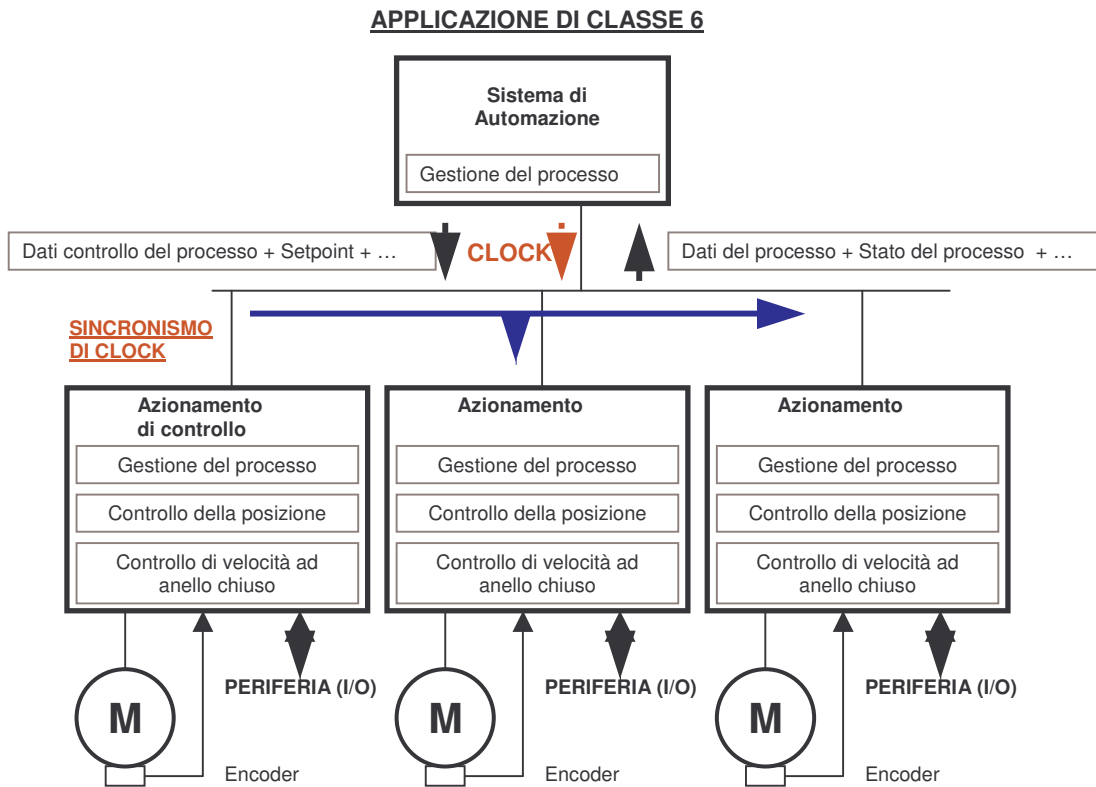



Figura 2.4 – Azionamento di classe 6

Per realizzare applicazioni come cambi elettronici, devono essere utilizzate comunicazioni slave to slave sincronizzate con il ciclo di clock.
Queste applicazioni sono normalmente realizzate con un azionamento di controllo e diversi azionamenti slave. Il termine azionamento di controllo significa, in questo contesto, che un azionamento, l’azionamento di controllo, fornisce informazioni sul processo, come ad esempio il valore attuale della posizione, agli altri azionamenti, gli azionamenti slave. In questo modo gli azionamenti slave coordinano all’istante il proprio movimento con le informazioni ricevute dall’azionamento di controllo.

2.3 SOLUZIONI EMERGENTI PER L'AUTOMAZIONE

Negli ultimi anni sono stati sviluppati diversi protocolli isocroni. Tutte le piu' grandi aziende per automazione cercano di imporre il proprio protocollo.
Verranno analizzati in dettaglio nei capitoli successivi i protocolli PROFIBUS DP-V2 e PROFINET.

2.3.1 PROFIBUS DP-V2

Lo versione isocrona di Profibus, PROFIBUS DP-V2 aggiunge nuove funzionalità al protocollo per il controllo tramite sincronismo di clock dei dispositivi slave da parte del master. Il controllo del sincronismo è indipendente dal carico del bus. Queste nuove funzionalità permettono processi di posizionamento molto precisi con deviazioni del ciclo di clock inferiori ad un microsecondo.
Tutti i dispositivi vengono sincronizzati dal master attraverso un messaggio broadcast di "global control" (GC). Un contatore speciale, "sign of life" permette di controllare la sincronizzazione. In figura con S1, S2, S3 è mostrato il tempo necessario per lo scambio dati ciclico di tre slave. Con Asinc. è mostrata la finestra temporale disponibile per le comunicazioni asincrone.

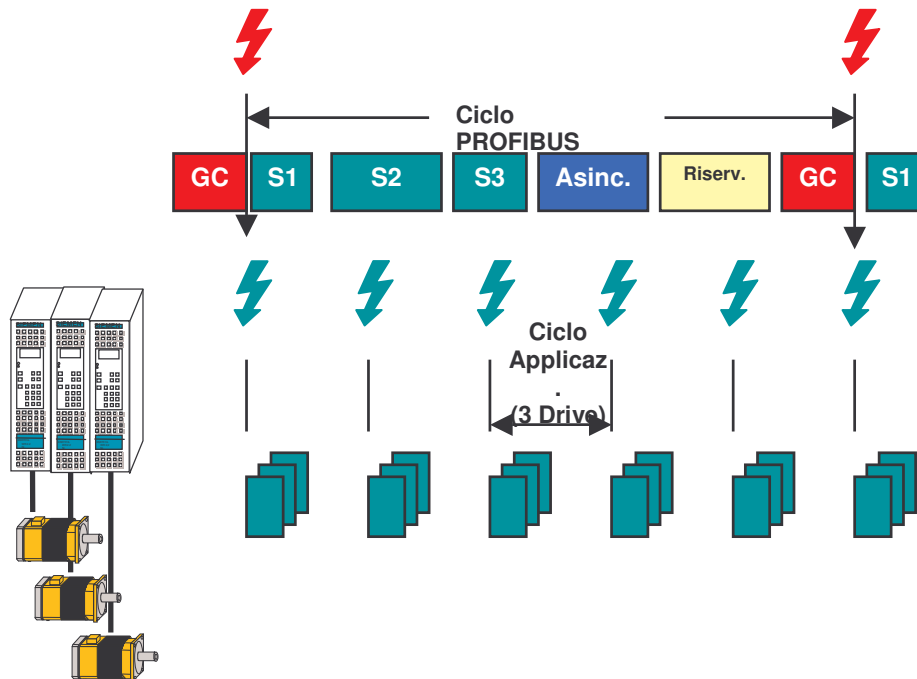



Figura 2.5 – Comunicazione isocrona PROFIBUS DP-V2

PROFIBUS DP-V2 implementa anche la comunicazione slave-to-slave per lo scambio dati fra gli slave. E' una comunicazione basata sul modello Publisher-Subscriber. Il modello è basato su un publisher (slave) che fornisce i suoi dati non solo al master DP, ma anche alle altre stazioni (subscriber), in modo che tutti gli altri slave possono accedere ai suoi dati ed agire di conseguenza. Tutto questo deve essere definito attraverso la configurazione del sistema PROFIBUS. In fase di configurazione si definiscono le relazioni slave-to-slave tra gli slave DP, i dati che il publisher rende disponibili, e a quali dati il subscriber deve accedere. La comunicazione slave-to-slave è associata allo scambio dati ciclico di PROFIBUS DP. La figura 2.6 mostra il meccanismo della comunicazione slave-to-slave.

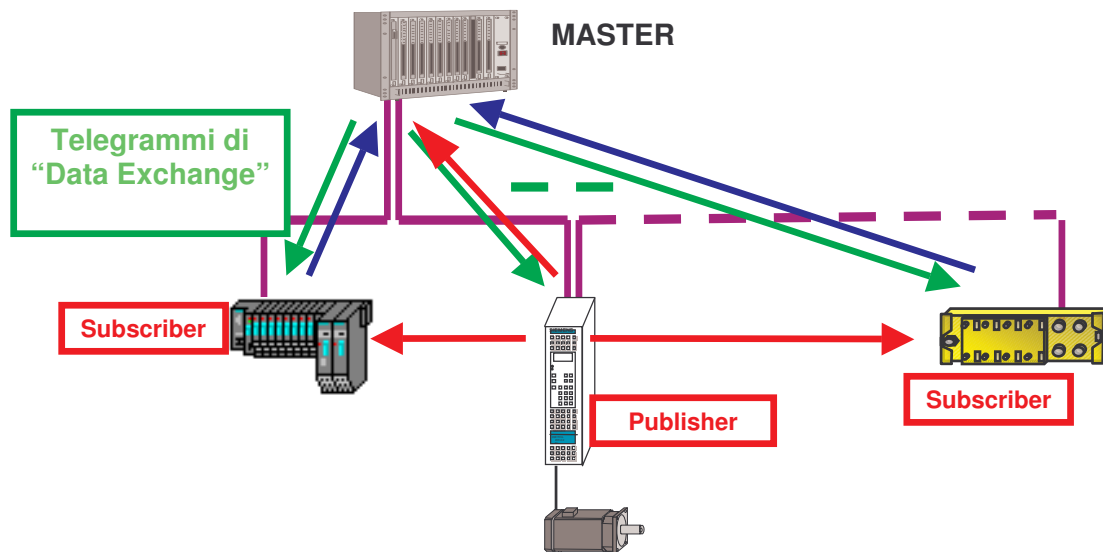



Figura 2.6 – Comunicazione slave to slave

2.3.2 ETHERNET A LIVELLO INDUSTRIALE

Nel settore dell'automazione industriale si sta cercando di introdurre la rete Ethernet anche a livello di campo. La rete Ethernet è molto diffusa a livello mondiale e l'introduzione di Ethernet a livello industriale può portare diversi vantaggi. Vantaggi economici perchè le cose diffuse costano poco. Un altro vantaggio consiste nella possibilità di utilizzare internet a livello industriale.
Ethernet esisteva già a livello industriale, ma solo fino a pochi anni fa era concentrata solo a livello amministrativo. Era la rete che collegava i vari uffici dell'amministrazione oppure collegava gli impianti di produzione a livello di area, non era la rete su cui viaggiavano i dati critici dell'automazione, con cui vengono, ad esempio, controllati gli azionamenti.
Con il termine Industrial Ethernet ci si riferisce in genere a quelle applicazioni che utilizzano Ethernet per le comunicazioni a livello di campo, ossia di PLC e periferia.
Per capire come è stato possibile utilizzare Ethernet a livello di bus di campo, si deve analizzare il suo funzionamento.

2.3.2.1 LA COMUNICAZIONE CSMA/CD

Ethernet usa come modello di comunicazione il tipo **CSMA/CD** (Carrier Sense Multiple Access with Collision Detection) le cui caratteristiche fondamentali sono:

- **Carrier Sense:** Tutte le stazioni Ethernet, prima di iniziare la trasmissione, sono obbligate a verificare che nessun altra stazione stia comunicando sulla rete.
- **Multiple Access:** Più di due stazioni possono trasmettere quando la rete è disponibile.
- **Collision Detection:** Determina il comportamento nel caso due sistemi trasmettano contemporaneamente.

Nel caso di collisioni, la trasmissione risulta disturbata e i dati sono di fatto inutilizzabili. Quando invia i dati, la stazione si assicura che il sistema destinatario

abbia ricevuto i dati corretti e, in caso di errore, presuppone che sia avvenuta una collisione. Tutte le stazioni interessate segnalano la collisione. Ogni stazione sospende la comunicazione per un periodo di tempo casuale dopo il quale ricomincia a trasmettere. Se sfortunatamente capita che i dati trasmessi collidano nuovamente, ogni stazione è obbligata a raddoppiare la durata dell'attesa casuale.
Queste situazioni sono definite collisioni multiple. Una collisione multipla avviene quando più di una collisione ha luogo nello stesso momento.
Le collisioni multiple sono il sintomo di problemi gravi. Le collisioni fanno invece parte delle comunicazioni su reti Ethernet e possono verificarsi di quando in quando.

2.3.2.2 LA STRUTTURA DEL PACCHETTO ETHERNET

La connessione tra le stazioni di comunicazione avviene in Ethernet attraverso la tecnica a "**commutazione di pacchetto**".
Ogni singolo pacchetto di dati può seguire un diverso percorso per raggiungere la destinazione finale. Il sistema ricevente riceve quindi spesso i dati in ordine sparso.
Ciò determina l'esigenza, da parte del sistema trasmittente, di etichettare ciascun pacchetto di dati con un numero che ne identifica la posizione all'interno della sequenza di trasmissione. Il sistema di ricezione può così utilizzare tali valori per ricostruire la sequenza originale dei dati.
Il vantaggio della tecnica di commutazione a pacchetto è che non necessita di una grande larghezza di banda. Essa è particolarmente utile quando occorre trasmettere i tradizionali dati di una rete (es. file di database) mentre si rivela inadeguata per la trasmissione di dati audio e video e in generale per le comunicazioni real-time.
Un pacchetto di dati Ethernet è un insieme di impulsi digitali trasmessi attraverso il mezzo trasmissivo.
Esso può essere di dimensione variabile (da 64 a 1518 byte) ed è composto da 4 parti principali :

- Preambolo: Le informazioni sono trasmesse in modo asincrono in forma di pacchetti. Non avendo una linea di clock, ma solo una linea dati è necessaria una fase di sincronizzazione. In questa fase si utilizzano i 7 byte del

preambolo, byte composti da bit 0 e 1 alternati che servono per sincronizzare il ricevitore.

- Inizio del Frame: Byte che delimita l'inizio del frame, uguale a 10101011
- Indirizzo destinatario: Contiene informazioni che riguardano il mittente del pacchetto (6 byte).
- Indirizzo sorgente: Contiene informazioni che riguardano il destinatario del pacchetto (6 byte). I valori, detti anche indirizzi MAC (Media Access Control) rappresentano gli indirizzi dei due sistemi collegati e rappresentano dei numeri univoci che identificano ciascun singolo dispositivo di rete.
- Lunghezza dei dati: indica il numero di byte presenti nel pacchetto dati (2 byte).
- Dati: Effettive informazioni che devono essere trasmesse (da 46 a 1500 byte).
- Checksum: Sequenza utilizzata per verificare la correttezza dei dati ricevuti. Il sistema trasmittente elabora questi byte mediante un algoritmo chiamato CRC(Cyclic Redundancy Check) (4 byte).

In figura è visibile il tipico messaggio ETHERNET:

Preamb.	Inizio Frame	Indir. Dest .	Indir. Sorg.	Lung. Dati/Tipo	Dati	Checksum
7 byte	1	6 byte	6 byte	2 byte	46<=N<=1500 byte	4 byte

Figura 2.7 – Struttura del pacchetto Ethernet

Ethernet è standardizzata tramite la norma IEEE 802.3.

Le specifiche coprono per esempio la tecnologia d'accesso, i metodi di trasmissione e i mezzi di trasmissione per Ethernet classica, per Ethernet veloce (100 Mbit/s).

Ethernet veloce a 100 Mbit/s è un'estensione compatibile di Ethernet di 10 Mbit/s ed è la versione più adatta per la comunicazione a livello di campo coniugando velocità e basso costo di componenti e infrastrutture.

2.3.2.3 IL MODELLO TCP/IP, UDP/IP

Un protocollo di comunicazione che è di fatto uno standard nel panorama dell'Information Technology (IT) è il protocollo TCP/IP. Vediamo in figura come si collocano i vari livelli in confronto ai livelli del modello ISO/OSI.

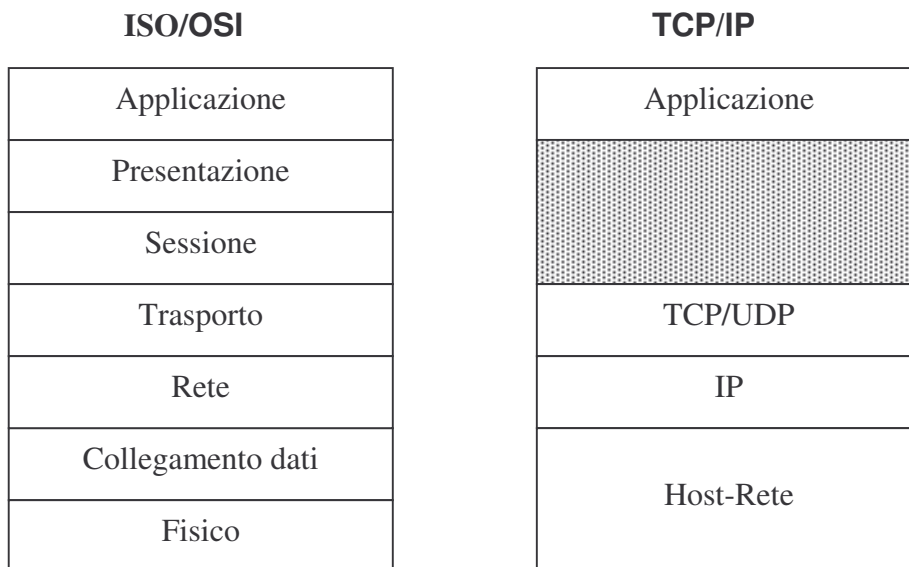



Figura 2.8 - Gli stack ISO/OSI e TCP/IP

I livello fisico e collegamento dati sono sostituiti dal livello Host-Rete. A livello 3, Rete, troviamo il livello IP, mentre a livello Trasporto possiamo trovare il livello TCP o il livello UDP. Per ragioni di efficienza non sono presenti i livelli Sessione e Presentazione. Troviamo infine il livello applicazione.

Il livello IP

La trasmissione con Internet Protocol software (IP) rappresenta una tecnica di commutazione a pacchetti "non afffidabile" o un servizio di datagram "non affidabile" fra una sorgente e una destinazione IP.

I datagrammi IP possono essere persi a causa di interferenza sul canale di trasmissione o per il sovraccarico della rete e possono arrivare parecchie volte o in un ordine differente rispetto a quello in cui sono stati trasmessi.

Può essere presupposto, tuttavia, che un datagramma se arrivato sia corretto.
Gli errori nel pacchetto possono essere identificati con una probabilità molto alta dal controllo totale dei 32 bit del pacchetto di Ethernet (cheksum).
Il protocollo IP è in grado di supportare anche livelli fisici diversi da ETHERNET quali per esempio linee telefoniche e fibre ottiche.
Il pacchetto IP (64 Kb) può essere frammentato in modo diverso a seconda del livello fisico con il quale dialoga.
Come è facile notare, se la dimensione massima del datagramma IP è di 65535 byte, per spedire un datagramma IP su una rete Ethernet lo si dovrà frammentare e i frammenti possono arrivare in ordine sparso. L'implementazione del protocollo IP richiede quindi molta memoria per il supporto della frammentazione.
Il datagramma IP è formato da un'header (intestazione) e da un campo dati.
La struttura dell'header IP è descritta in figura.

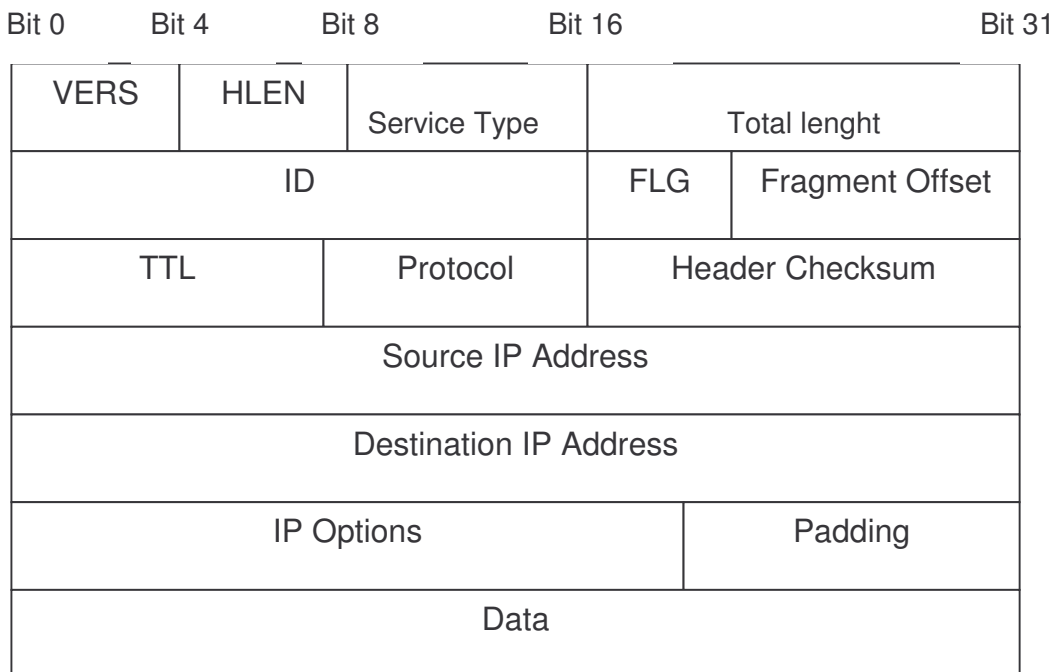



Figura 2.9 - Struttura dell'header IP

Il livello TCP

TCP garantisce che i dati siano trasmessi dal mittente al ricevente senza errore, nella sequenza corretta e completamente.

Il TCP è connection-oriented, cioè un collegamento è installato tra due stazioni prima della trasmissione ed è chiuso dopo la trasmissione.

Il TCP funziona come segue:

Un canale FDX (duplex pieno) è installato fra due programmi di applicazione in varie stazioni della rete. Questo canale consente la trasmissione simultanea ed indipendente dei flussi di dati in entrambi i sensi, senza interpretazione.

Il TCP ha meccanismi per il controllo continuo di una connessione stabilita. Gli errori nel trasferimento dei dati, per esempio una rottura inattesa nel collegamento, una congestione nella rete ecc., sono segnalati al software di applicazione.

Il software di applicazione dà i relativi dati al TCP in segmenti di dimensione casuale e in intervalli di tempo di lunghezza casuale. TCP/IP conserva i segmenti fino a che non possano essere trasmessi, li divide ove necessario in blocchi più piccoli per soddisfare il sistema di trasmissione e ricrea i dati corretti che fluiscono al ricevente.

La struttura dell'header TCP, che può avere lunghezza variabile 20-60 byte è descritta in figura.

Bit 0 — Bit 18 — Bit 31

<table>
<tr><td colspan="8">Source Port</td><td colspan="1">Destination Port</td></tr>
<tr><td colspan="9">Sequence Number</td></tr>
<tr><td colspan="9">Acknowledgment Number</td></tr>
<tr><td>Data Offset</td><td>Reset</td><td>U R G</td><td>A C K</td><td>P S H</td><td>R S T</td><td>S Y N</td><td>F I N</td><td>Window</td></tr>
<tr><td colspan="8">Checksum</td><td>Urgent Pointer</td></tr>
<tr><td colspan="9">Option Padding</td></tr>
<tr><td colspan="9">Data Bytes</td></tr>
</table>

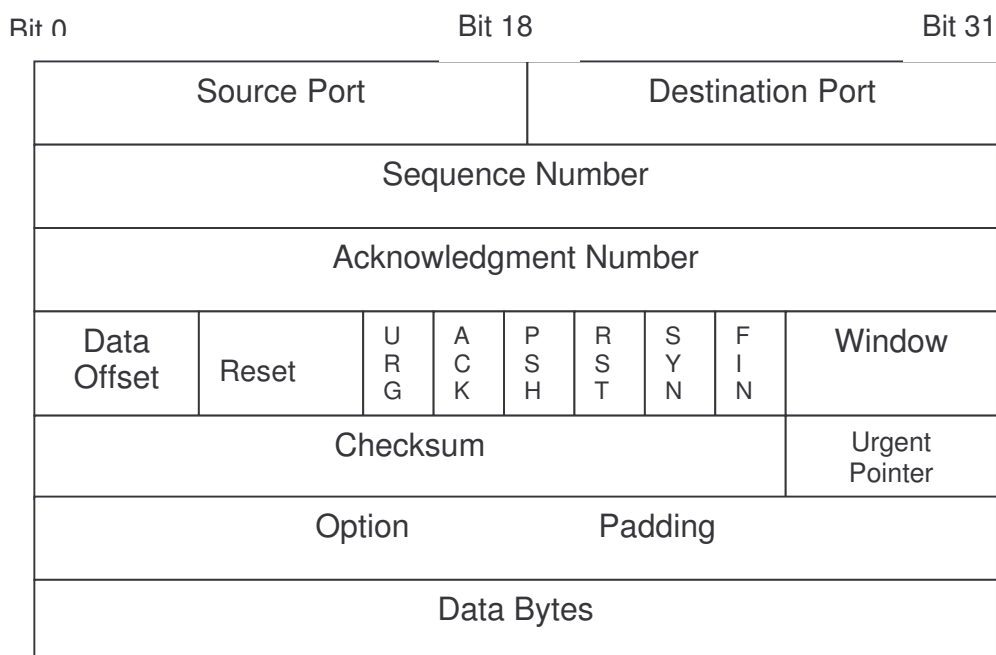



Figura 2.10 - - Struttura dell'header TCP

Il livello UDP

In sostituzione al protocollo TCP/IP talvolta viene utilizzato il protocollo UDP/IP.
Si tratta di un protocollo non connection-oriented che, se da un lato ha il vantaggio di essere più veloce, dall'altro paga questa velocità con un livello di sicurezza minore rispetto a quello garantito dal protocollo TCP/IP.
La struttura dell'header UDP è descritta in figura.

Bit 0   Bit 31

UDP length	Destination Port
Source Port	UDP checksum

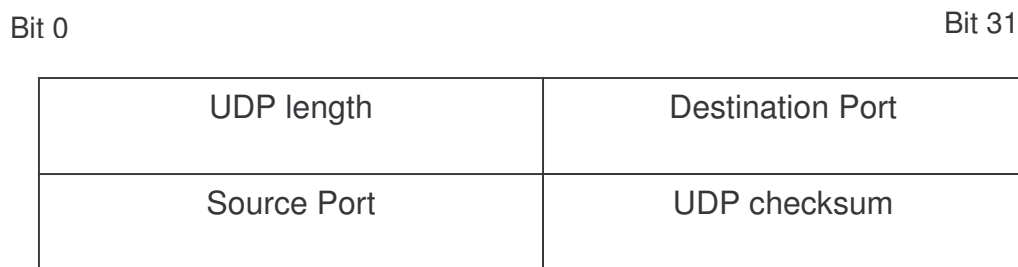



Figura 2.11 - Struttura dell'header UDP

L'UDP pur non supportando meccanismi di sicurezza, è molto più interessante per applicazioni a livello di campo grazie alla semplicità ed economicità dell'implementazione.

Il livello applicazione

Contiene i protocolli di alto livello che vengono usati nelle applicazioni quali:

Telnet : Terminare Virtuale
FTP: Trasferimento file
SMTP e POP: posta elettronica
DNS: mapping fra nomi di host e indirizzi IP
NNTP: newsgroup
HTTP: Word Wide Web
Questi sono i tipici protocolli di internet basati su TCP/IP che tuttavia risultano di limitato utilizzo a livello industriale e praticamente inapplicabili a livello di campo se non per funzioni diagnostiche a bassissima priorità.

2.3.3 ETHERNET COME BUS DI CAMPO

Ipotizzando una rete 10Base-T, si osserva che per sensori o attuatori semplici, quale ad esempio un sensore di umidità che si limita a trasmettere due bytes (1 dato = 2 bytes = percentuale di umidità), Ethernet risulta poco efficiente. Infatti il tempo T_{Eth} per trasmettere l'informazione risulta pari a 67.2 μs con una efficienza η_{ETH} risultante inferiore al 10%.

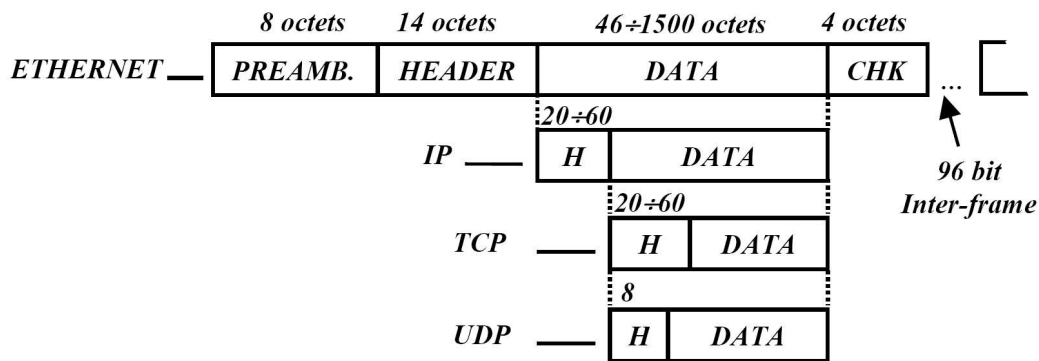



Figura 2.12

$$T_{Eth} = (64+8+12)\times 8\times Tbit = 67.2\,\mu\text{s} \qquad \eta_{Eth} = \frac{d\times 8\times Tbit}{(64+8+12)\times 8\times Tbit} \approx \begin{cases} 2.4\%\,(d=2) \\ 9.6\%\,(d=8) \end{cases}$$

Ovviamente se si considerano i protocolli a livello superiore, come IP o TCP, l'efficienza tende a peggiorare, pur considerando l'header (H) minimo, e la rete libera (il data-rate può essere notevolmente ridotto in caso di rete affollata).

$$T_{IP} = (64+8+12)\times 8\times Tbit = 67.2\,\mu\text{s} \qquad \eta_{IP} = \frac{d\times 8\times Tbit}{(64+8+12)\times 8\times Tbit} \approx \begin{cases} 2.4\%\,(d=2) \\ 9.6\%\,(d=8) \end{cases}$$

$$T_{TCP} = \begin{cases} (64+8+12)\times 8\times Tbit = 67.2\mu s\,(d=2) \\ (66+8+12)\times 8\times Tbit = 68.8\mu s\,(d=8) \end{cases} \qquad \eta_{TCP} = \frac{d\times 8\times Tbit}{(64/6+8+12)\times 8\times Tbit} \approx \begin{cases} 2.4\%\,(d=2) \\ 9.3\%\,(d=8) \end{cases}$$

Storicamente l'impiego di Ethernet come bus di campo è risultato penalizzato dal non-determinismo e, per quanto concerne la scarna messaggistica dei sensori e degli attuatori, dalla bassa efficienza, dato che il numero di byte trasmessi è elevato rispetto al numero di byte che contengono effettivamente informazione. Oggi, la diffusa presenza di switch, che inoltrano il messaggio solo al destinatario, e l'aumento della velocità di trasmissione (10MHz, 100MHz, 1GHz) riducono

notevolmente la probabilità di collisione, per cui Ethernet diventa una soluzione sempre più interessante anche per le applicazioni veloci quali i bus di campo.
Il non determinismo di Ethernet non sembra particolarmente penalizzante rispetto ai bus di campo nei sistemi a bassa dinamica, in quanto una rete in ambiente industriale viene progettata con un ridotto numero di utenti e, in genere, segmentata su più reti locali. Anche la bassa efficienza, aggravata dalla presenza di protocolli di una certa complessità quali UDP-IP o TCP-IP, risulta meno penalizzante grazie alle elevate velocità di trasmissione, e in generale è un argomento meno valido rispetto alla presenza delle infrastrutture. Ethernet ha la sua naturale evoluzione in Internet, che consente una diagnostica decentrata di semplice implementazione. E' per questo motivo che anche a livello industriale vanno affermandosi protocolli standard, quali quelli utilizzati dai comuni browser, come Internet Explorer o Netscape Navigator (IP, TCP, HTTP) e cominciano a comparire sul mercato i primi web-sensors, ossia sensori direttamente interfacciati su Internet e consultabili mediante un comune browser anche se non pienamente compatibili ai protocolli in questione.

2.3.3.4 IL COLLEGAMENTO DEI DISPOSITIVI TRAMITE SWITCH

La rete Ethernet che funziona utilizzando il protocollo CSMA/CD è adatta per un bus di campo?
L'utilizzo di CSMA/CD può generare delle collisioni fra i pacchetti, quindi sembrerebbe non adatta per applicazioni industriali a livello di campo.
Il problema delle collisioni è stato risolto connettendo i vari dispositivi tramite uno switch. Gli switch inoltrano il messaggio solo al destinatario.
In questo caso non abbiamo più dispositivi connessi su uno stesso bus. Ogni dispositivo è connesso ad una porta dello switch. Se la stazione A vuole parlare con la stazione B e la stazione C vuole parlare con la stazione D si creano all'interno dello switch dei circuiti per cui le informazioni passano direttamente da C e D e da A a B.
Cosa succede se la stazione A sta parlando con la stazione B, ma anche D vuole parlare con B? Non abbiamo collisione perché lo switch è un elemento con memoria.

Può memorizzare quello che la stazione D voleva dire alla stazione B e spedirlo quando la stazione B ha finito di parlare con A. All'interno dello switch ci sono dei buffer che permettono di memorizzare i vari pacchetti nel caso la destinazione sia occupata.
Questo mi permette, fino a quando non saturo il buffer, di non perdere nessun pacchetto. Il costo di uno switch è in genere proporzionale alla dimensione di questi buffer.

2.3.3.5 INCAPSULAMENTO

L'idea più semplice per utilizzare ethernet con i protocolli di bus di campo esistenti è stata quella di incapsulare i protocolli esistenti all'interno del campo dati del pacchetto ethernet.
Un esempio di incapsulamento è **Ethernet /Ip** (Ethernet Industrial Protocoll).
Si è incapsulato all'interno del pacchetto ethernet un bus di campo quale DeviceNet. DeviceNet in questo caso diventa uno dei tanti protocolli applicativi oltre a Http e Ftp. Sotto il livello applicativo ci si avvale ancora del protocollo TCP/IP.
EtherNet/IP mantiene invariati il livello fisico ed il metodo di accesso CSMA/CD Standard Ethernet IEEE, utilizza i servizi offerti dai protocolli TCP/IP e integra tutte le funzioni del protocollo applicativo CIP - Control and Information Protocol, cuore della tecnologia di comunicazione NetLinx di Rockwell Automation. L'impiego di EtherNet/IP permette in pratica di realizzare soluzioni di controllo distribuite con prestazioni paragonabili a quelle di ControlNet.
Un altro degli standard più diffusi, grazie alla sua semplicità è **Modbus** over TCP.
In tutti questi standard si tende a preservare il livello applicazione rispetto al bus di campo adottato. Si tratta di protocolli in genere poco adatti ad applicazioni di "motion control" e i tempi di ciclo sono nell'ordine della decina di ms.

Un approccio di questo tipo presenta i seguenti svantaggi. Devo passare attraverso strati software: prima attraverso l'incapsulamento, poi attraverso TCP/IP , ecc. e tutto questo crea dei ritardi che oltretutto non sono costanti.

Sono quindi stati sviluppati altri protocolli per applicazioni in tempo reale, i cosiddetti protocolli isocroni real-time.

2.3.4 PROFINET

Particolarmente completo risulta l'approccio di **PROFINET** che introduce il concetto di oggetti per l'automazione. Come protocollo di comunicazione PROFINET presenta due diverse tipologie di prodotto per coprire le molteplici esigenze: un protocollo (PROFINET CBA) poco adatto al tempo reale ma basato su standard di accesso molto diffusi quali RPC/DCOM/OPC, e un protocollo pensato per il campo e il tempo reale (PROFINET IO).
A livello applicativo PROFINET CBA introduce il concetto di modulo tecnologico.
La funzione di un impianto automatico o di una macchina in un processo di manufacturing è effettuato con l'interazione tra meccanica, parte elettrico/elettronica e parte di controllo logico/software. Lavorando a questo principio, PROFINET CBA definisce le parti "meccaniche", "elettriche/elettronica" e "logica di controllo/software" in termini funzionali per formare un unico modulo tecnologico.
Un modulo tecnologico permette quindi di fornire al progettista esattamente l'hardware e il software del quale ha bisogno, visto come un singolo oggetto, il Componente per l'Automazione PROFINET.

Figura 2.13 - Modello componente in PROFINET

I componenti vengono descritti con file XML che hanno la stessa funzione dei file GSD di PROFIBUS: permettere l'integrazione di dispositivi di diversi produttori. Creare una descrizione standard di un componente secondo delle regole comuni svincola il progettista dalla conoscenza di tool di programmazione, configurazione e parametrizzazione che possono essere diversi per ogni componente. Tutti i componenti sono utilizzabili in modo identico senza bisogno di conoscenze a priori.

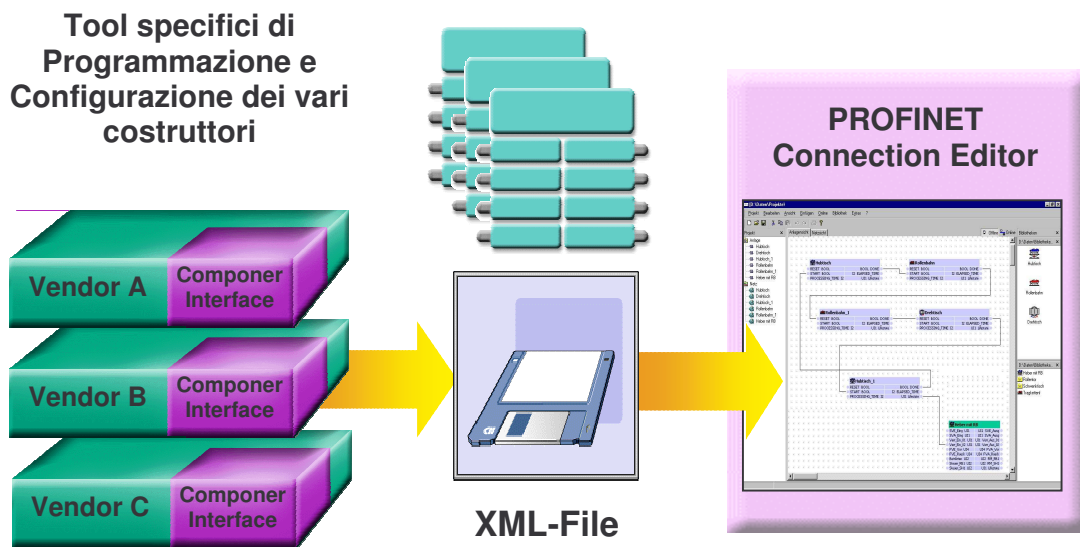



Figura 2.14

Al progettista non resta che collegare tra di loro i vari componenti per mezzo di tool grafici per ottenere il progetto completo. Da notare che praticamente non è richiesta nessuna conoscenza del mezzo fisico su cui viaggiano le informazioni, per cui PROFINET CBA è virtualmente aperto a ogni tipo di protocollo a patto che i componenti siano rappresentati secondo le specifiche PROFINET.

Per esempio, grazie a degli accoppiatori attivi detti proxy, PROFINET CBA integra Profibus. Il proxy è il rappresentante di ciascuno dei dispositivi collegati al bus di campo per quanto riguarda il mondo PROFINET. Il proxy si deve occupare di gestire la comunicazione verso il Ethernet e verso il bus di campo, cioè si comporta da gateway per quanto riguarda la comunicazione fisica. Inoltre il proxy è un oggetto che sostituisce i dispositivi nel senso che esso realizza la vista PROFINET CBA (ad

alto livello) del dispositivo di campo rendendola disponibile al mondo esterno. Rispetto al bus di campo il proxy svolge le funzioni di un master. Come conseguenza dell'introduzione del proxy, tutti i dispositivi del bus di campo organizzati dietro esso sono presentati come nodi indipendenti di PROFINET (oggetti) e tali componenti risultano indistinguibili dai componenti posti direttamente su Ethernet. Importante sottolineare che tutte le configurazioni impostate a livello PROFINET CBA per gli oggetti su bus di campo sono salvate nel proxy e non nei dispositivi di campo.

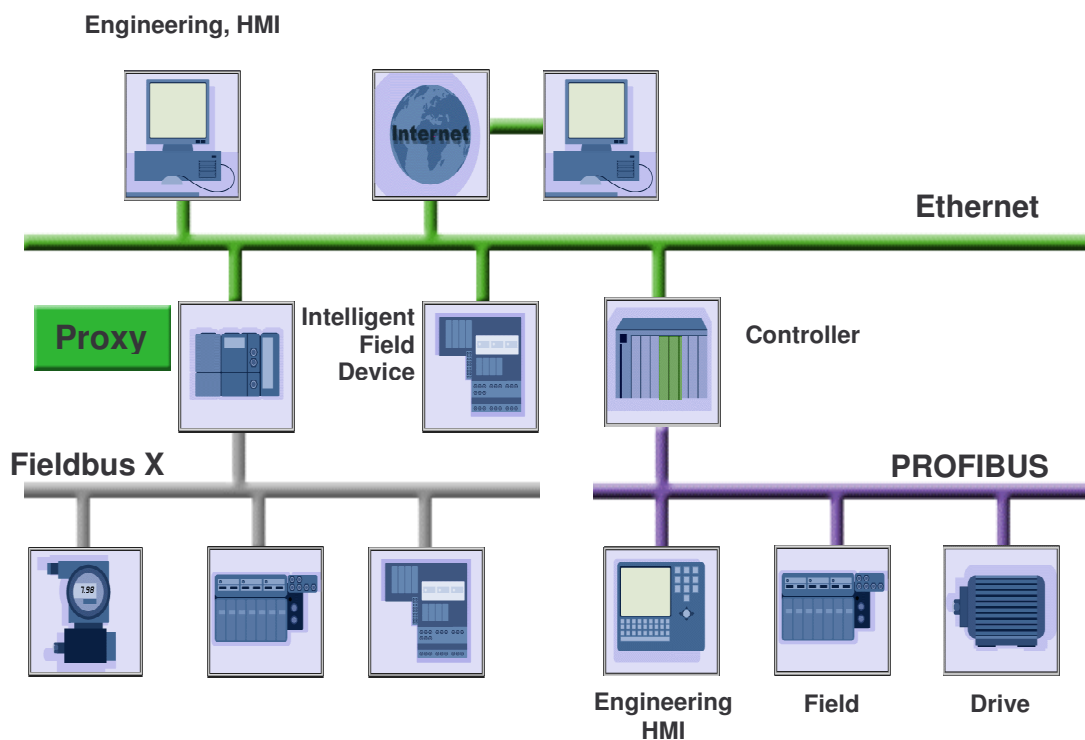



Figura 2.15 – Collegamento tramite proxy

Data la difficoltà di realizzare interfacce OPC residenti, il dispositivo con interfaccia PROFINET CBA non implementa un server OPC direttamente sul dispositivo di campo ma solo una parte. In particolare viene implementato il livello ACCO, che scambia dati tramite DCE-RPC (Remote Procedure Call, protocollo basato su TCP/IP) con una stazione client più potente (es. PC) dove è implementato anche la parte mancante del server. Lo scambio dati tra dispositivi non passa però tramite

OPC ma è direttamente a livello ACCO in modo da garantire performance migliori, che comunque rimangono lontane dal real-time.

Si passa ora a parlare dei protocolli di rete usati da PROFINET CBA e da PROFINET IO. Le esigenze dei due protocolli sono differenti e quindi ovvio che lo stack di comunicazione sia complesso (vedi figura seguente). In generale PROFINET definisce tre tipi di modalità di comunicazione: Non Real-Time (Non RT), Real-Time (RT) o Soft Real Time (SRT), Real Time Isocrono (IRT Isochronous Real Time).

In generale PROFINET CBA utilizzerà il primo modo di comunicazione, mentre PROFINET IO usa le due classi di comunicazione real-time. Il canale Real-time permette la trasmissione di dati ciclici ed aciclici.

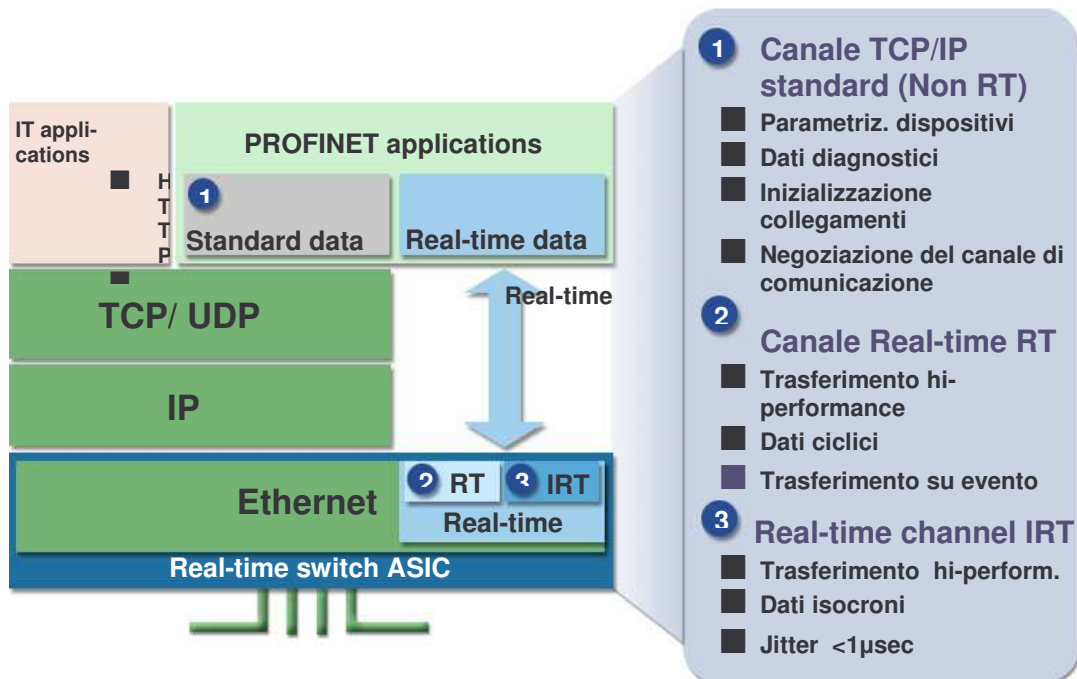



Figura 2.16 – Lo stack di PROFINET

Il grafico sottostante evidenzia la distribuzione dei tempi di risposta dei tre diversi approcci PROFINET alla comunicazione.

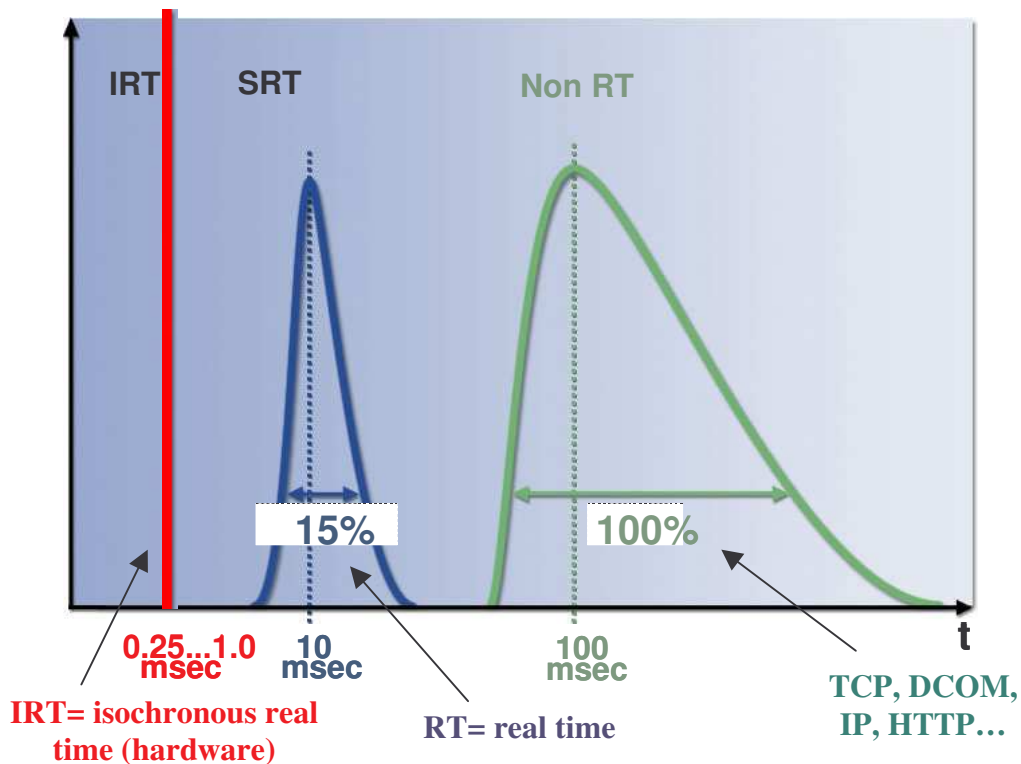



Figura 2.17 – Tempi di risposta

La comunicazione standard attraverso il protocollo TCP\IP fornisce tempi di risposta medi attorno ai 100 ms. RT (SRT) evidenzia un miglioramento dei tempi di trasmissione di un fattore almeno 10 e i tempi di risposta si attestano infatti tra gli 1 e i 10 ms. Nel caso della comunicazione mediante Isochronous Realtime IRT si ottengono addirittura tempi di risposta costantemente attestati attorno a 1 ms. Anche la varianza subisce una riduzione progressiva il che si traduce ovviamente in una minor oscillazione dei tempi di risposta (jitter) attorno al valore medio della distribuzione.

Per potere raggiungere il risultato dichiarato dai dispositivi che usano RT si assegna ai pacchetti in PROFINET IO una priorità alta in conformità con gli standard IEEE802.1q e IEEE802.1p: i dati che fluiscono fra i dispositivi sono controllati dai componenti della rete (switch con capacità VLAN) in base a questo concetto di priorità.

Per quanto riguarda il secondo approccio in tempo reale, l'Isochronous RealTime (IRT) si tratta in questo caso di una soluzione hardware, che prevede l'uso di

particolari ASIC, con funzionalità di switch, che permettono di operare separando nel tempo la comunicazione RT e Non RT da quella IRT. Ogni partecipante alla comunicazione è sincronizzato con tutti gli altri (tramite IEEE1588) e l'accesso al mezzo fisico è fatto in divisione di tempo. In questo modo è come se si avessero a disposizione dei canali separati, uno per il traffico non critico e uno per IRT.

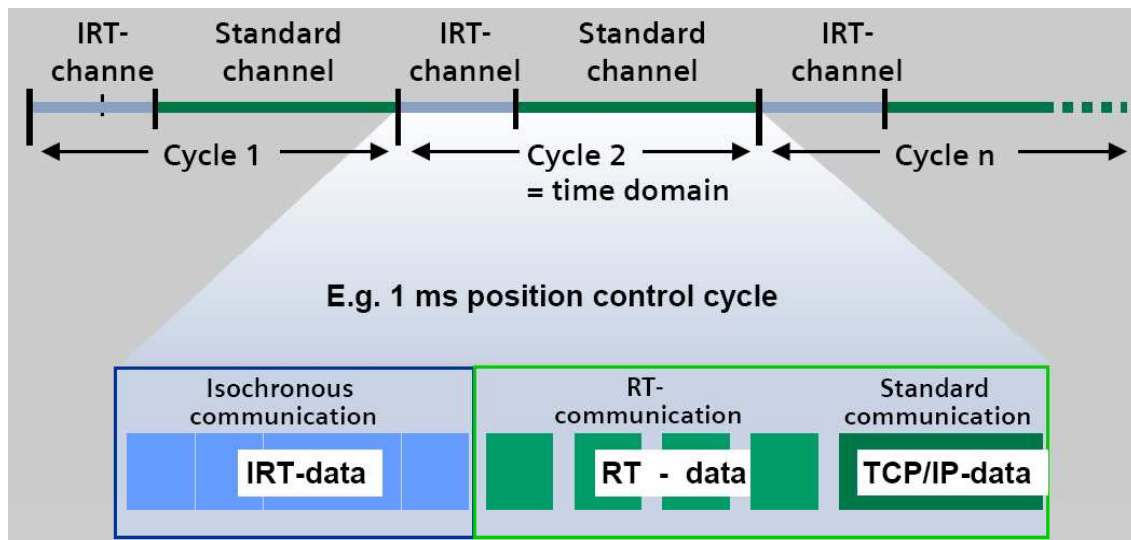



Figura 2.18 – Distribuzione temporale della comunicazione

Da notare che non ci sono messaggi di sincronizzazione espliciti perché l'istante di inizio ciclo è conosciuto da tutti, visto che i componenti sono sincronizzati.

Tutto questo è possibile in PROFINET IO perchè anche la trasmissione dei dati nella rete è ottimizzata. Le misure indicano infatti che i tempi di trasmissione fra due stazioni su Ethernet possono ammontare fino a 20ms nelle reti commutate tramite switch normali se si verifica un carico molto alto della rete: occorre usare degli switch differenti. Gli ASIC della serie ERTEC prodotti da SIEMENS e NEC possono capire in che fase del ciclo si trova il bus Ethernet e decidere quale tipo di traffico far passare. In figura 2.26 è mostrato il funzionamento di uno switch contenente un ASIC della serie ERTEC. Allo switch arrivano sia i pacchetti isocroni real-time IRT sia i pacchetti non IRT.

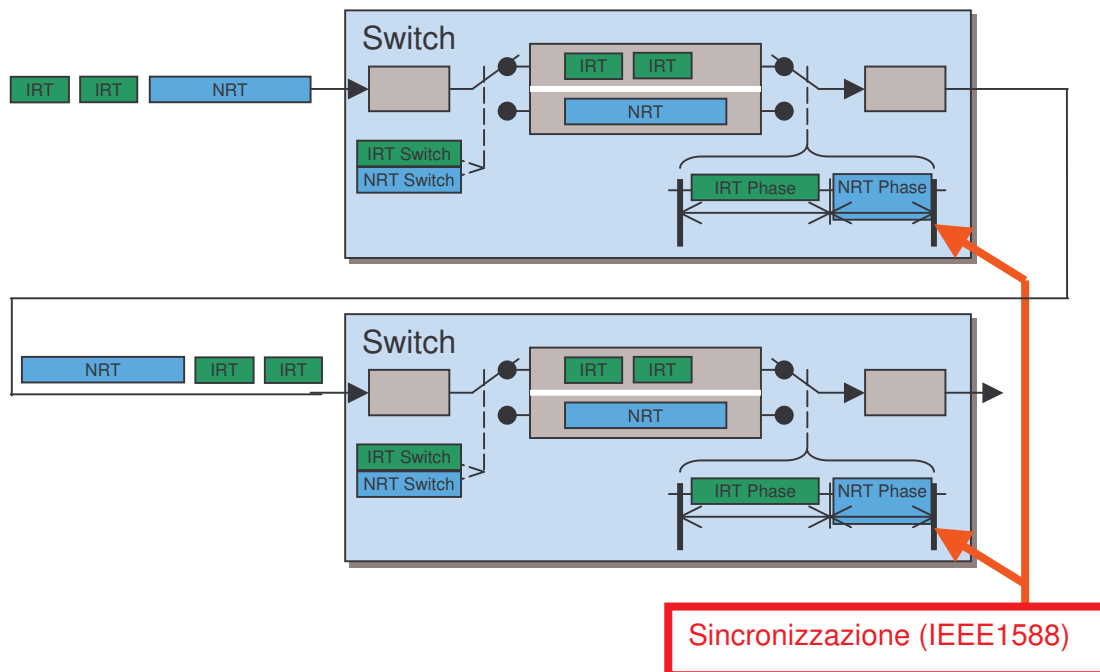



Figura 2.19 – Principio di funzionamento dello switch IRT

Se ci si trova nella fase real-time i pacchetti IRT hanno la precedenza mentre i pacchetti "Non IRT" vengono bufferizzati. Siccome anche questi switch sono sincronizzati, durante la fase IRT si effettua un inoltro dei pacchetti su base temporale e non sulla base dell'indirizzo MAC eliminando quindi ogni minimo tempo di ritardo.

2.4 LA TECNOLOGIA DELLA TRASMISSIONE ETHERNET

Lo standard internazionale ISO/IEC 11801 e la sua equivalente europea EN 50173 definiscono una rete standard IT.

I due standard sono in gran parte identici nel contenuto.

Né l'uno né l'altro standard tengono conto dei requisiti specifici che devono essere soddisfatti da reti Ethernet in un ambiente industriale come per esempio:

- Il percorso del cavo d'installazione
- Il livello specifico di rete per ogni machina/impianto
- La topologia della rete
- I cavi ed i connettori robusti e compatibili all'ambiente industriale progettati per fare fronte alle richieste speciali per quanto riguarda la temperatura, l'umidità, la polvere e la vibrazione.

"La tecnologia della trasmissione e del cablaggio di PROFINET " definisce così un sistema di cablaggio industriale per "Ethernet veloce" basato sui requisiti fondamentali dell' IEC 11801.

2.4.1 LE TOPOLOGIE DELLA RETE

Le topologie della rete più frequenti includono le strutture a stella, a linea, ad albero e ad anello. In pratica un sistema consiste di una miscela delle strutture che sono descritte più nei particolari qui sotto.

Queste strutture possono essere implementate attraverso una coppia di cavi in rame o attraverso fibre ottiche.

Topologia a stella

La struttura a stella è caratterizzata da un distributore centrale del segnale (switch) con i diversi collegamenti ai terminali di tutta la rete.

Le applicazioni per le strutture della rete a forma di stella sono zone ad alta densità di dispositivi e corto raggio di espansione, per esempio piccole cellule di manufacturing o una singola macchina di produzione.

Topologia ad albero

La topologia ad albero deriva dall' unire insieme parecchie reti stella per formare una rete. Una miscela di fibra ottica e di cablaggio di doppini twistati è possibile ove necessario.

Tale topologia è utilizzata per la divisione di installazioni complesse in sotto installazioni di minor complessità.

Topologia a linea

La struttura a linea è realizzata per mezzo di uno switch vicino ad ogni dispositivo da collegare o da uno switch integrato in ogni dispositivo.

È la struttura più utilizzata nel settore dei bus di campo perché è la più economica dal punto di vista dell'istallazione dei cavi.

Topologia ad Anello (ridondanza)

Chiude le estremità di una dorsale con un collegamento supplementare e produce una struttura d'anello.

Le topologie d'anello sono usate nei sistemi con alti requisiti di disponibilità per proteggere la rete da linee danneggiate o componenti difettosi.

Capitolo 3

PROFIBUS DP-V2

3.1 PROFIBUS DP-V2

Nei paragrafi seguenti verranno descritte in dettaglio le caratteristiche della versione 2 di PROFIBUS. Le funzionalità di PROFIBUS DP-V2 sono state definite come un estensione della norma principale IEC 61158 in supporto alle esigenze di controllo del movimento. DP-V2 è un estensione delle funzionalità di DP-V1. Per una descrizione delle funzioni base del protocolli DP-V0 e DP-V1 si rimanda all'appendice B. Vediamo alcuni esempi di funzionamento nel campo degli azionamenti, campo per il quale è stato sviluppato.

3.1.1 RETE CON FUNZIONALITÀ DI BASE PROFIBUS DP (DP-V0, DP-V1)

Sulla rete PROFIBUS si possono connettere fino a 126 dispositivi, master o slave, in una configurazione a singolo master oppure a più master. In questo modo, differenti dispositivi, come azionamenti o periferiche di I/O, possono operare sullo stesso bus. Possiamo catalogare i dispositivi come:

- Master DP di Classe 1 (DPM1)	PLC
- Master DP di Classe 2 (DPM2)	Dispositivi per la programmazione, dispositivi per la configurazione, pannelli operatori

- Slave DP Dispositivi periferici (I/O, azionamenti, attuatori/sensori)

Nel caso di configurazioni multimaster, alcuni master DPM1 possono costituire dei sottosistemi che sono indipendenti uno con l'altro (ogni master ha un certo numero di slave assegnati a se stesso). Oltre a questi dispositivi ci possono essere sul bus master DPM2. Il processo di scambio del Token assicura l'autorizzazione per l'accesso al bus ai vari master all'interno di una finestra temporale ben definita.

Attraverso il meccanismo di global control, un gruppo di slave o tutti gli slave possono essere sincronizzati con un evento di controllo (SYNC = "attivazione simultanea" dei valori di setpoint; FREEZE = "congelamento" simultaneo dei valori attuali) (Appendice B, § B.4.11). *Non è tuttavia possibile uno scambio di dati realmente isocrono, ciò è dovuto a ripetizioni di telegrammi causati da interferenze o dai meccanismi di diagnostica inseriti durante lo scambio dati ciclico.*

Nella figura 3.1 è mostrata una tipica rete PROFIBUS DP contenente delle periferiche di I/O e degli azionamenti di classe 1 e di classe 3 (§ 2.3) che non necessitano per il funzionamento uno scambio di dati isocrono e un dispositivo DPM2.

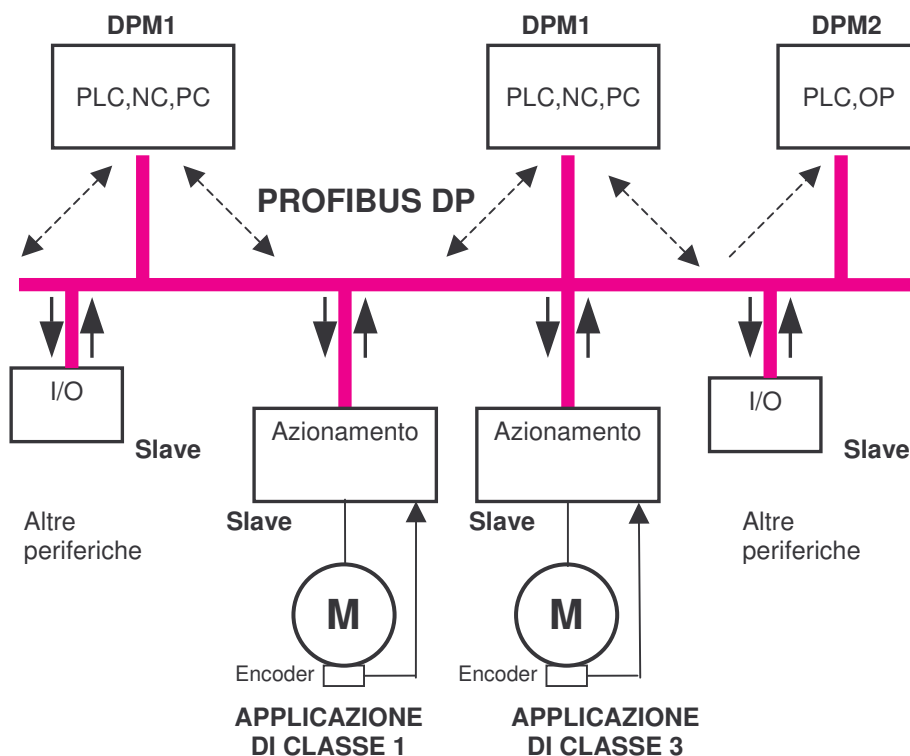



Figura 3.1 – Tipica rete PROFIBUS DP

3.1.2 RETE ISOCRONA MONOMASTER

Si utilizza questo tipo di configurazione quando gli azionamenti richiedono un alto livello di sincronizzazione. In figura si possono notare degli azionamenti di classe 4, oltre a quelli di classe 1 e 3, che richiedono una trasmissione dati sincronizzata al ciclo di clock. Il master DPM1 passa, tramite un telegramma di global control trasmesso in modo isocrono, l'informazione del ciclo di clock agli slave che si sincronizzano su questo impulso.

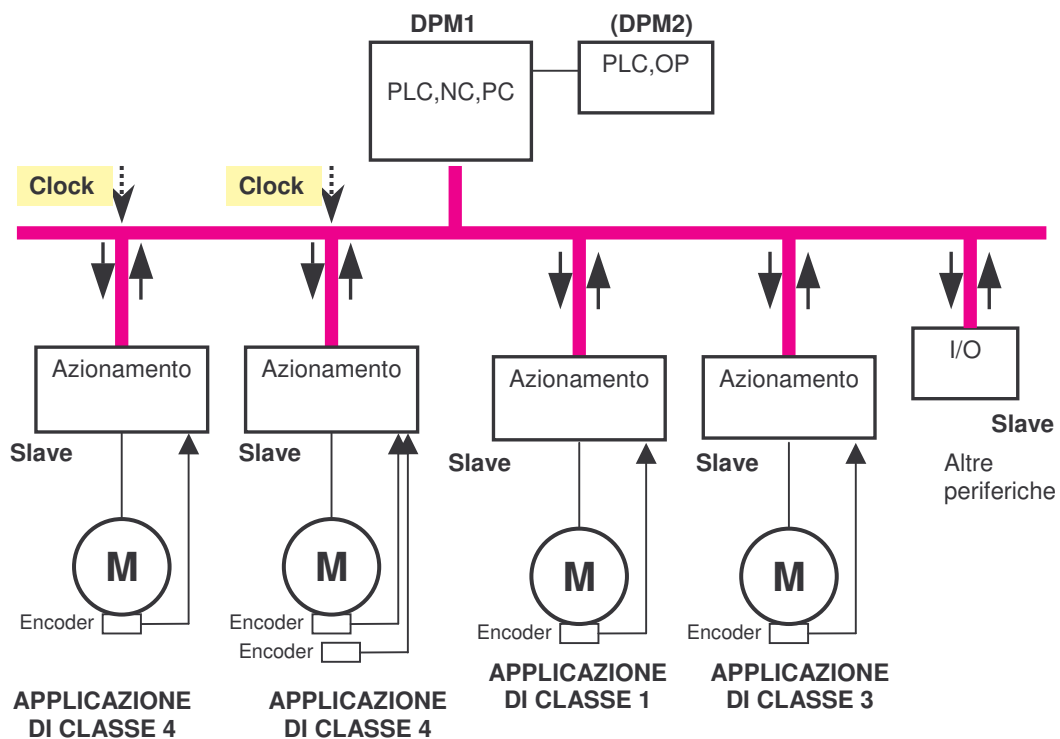



Figura 3.2 – Rete isocrona monomaster

Le temporizzazioni per il trasferimenti dei setpoint e dei valori acquisiti quanto i controlli effettuati dal master di livello più alto possono essere parametrizzate. I parametri delle temporizzazioni fanno riferimento all'impulso di clock.

Nel ciclo DP, deve esserci sufficiente tempo per i seguenti elementi:

- scambio dati ciclico (Data_Exchange, DP-V0) con tutti gli slave presenti sul bus
- un canale di comunicazione aciclica (DP-V1)
- ripetizioni del telegramma
- richieste diagnostiche

Un pannello operatore DPM2 deve comunicare con gli slave tramite un master DPM1 (Routing).

3.1.3 RETE ISOCRONA MULTIMASTER

Il tipo di generazione dell'impulso di clock permette l'utilizzo sul bus di un solo master DP di classe 1 sincrono. Non possono coesistere due master che generano entrambi degli impulsi di clock tramite telegrammi broadcast. Se esistono altri master (DPM1, DPM2) sullo stesso bus questi devono essere master DP non sincroni. La presenza di master non sincroni ha come effetto un aumento del tempo di ciclo DP minimo, T_{DP}.

Quando vengono utilizzate configurazioni isocrone multimaster sono raccomandate le seguenti restrizioni:

- un master di classe 1 può usare senza restrizioni tutti i servizi disponibili; altri dispositivi di comunicazione sul bus devono limitare a 2 il numero di canali aciclici di classe 2
- un master DPM2 deve essere certificato per applicazioni con ciclo di clock sincronizzato

3.1.4 MODO ISOCRONO (IsoM)

La figura 3.3 seguente mostra la sequenza di un ciclo DP isocrono.

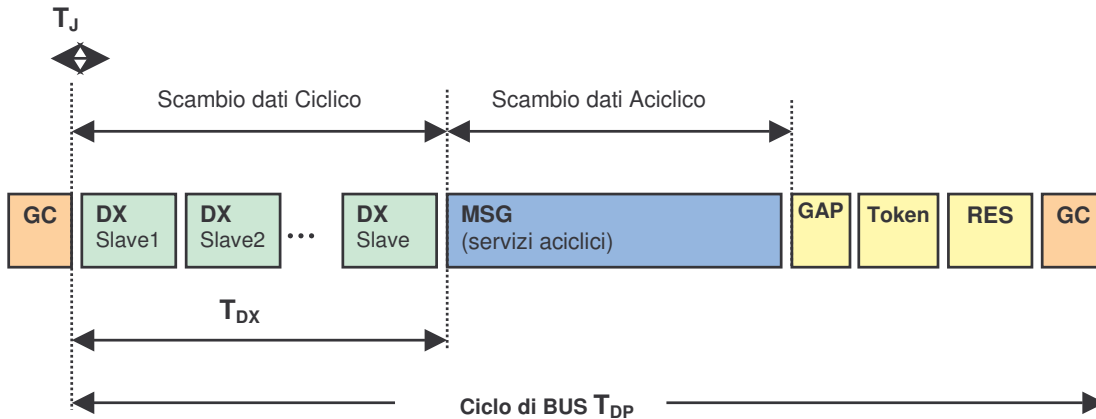



Figura 3.3 – Ciclo DP isocrono

Ogni ciclo di bus inizia con un telegramma di Global Control trasmesso in broadcast. Nel telegramma di Global Control (GC) il bit "Clock" è settato a 1. Questa informazione di clock è usata dagli slave per sincronizzarsi con il ciclo di clock del bus.

Il master inizia a condurre lo scambio di dati ciclico con gli slave (MS0).

Dopo di questo avviene uno scambio di dati aciclico (MS1).

Segue il GAP Update Time (Appendice B), durante il quale vengono inviate FDL_Request per la ricerca di nuove stazioni attive, e un telegramma di Token.

Alla fine trova posto una finestra temporale parametrizzabile riservata, che è normalmente riempita con FDL_request al fine di realizzare un ciclo di bus costante.

Il nuovo ciclo di bus inizia di nuovo con un telegramma di GC.

Nel telegramma di Global Control può venir settata la combinazione di bit per le funzionalità di SYNC e FREEZE. Tuttavia, questo ha solo interesse per gli slave che non sopportano la modalità isocrona.

T_J (Tempo di Jitter)

TJ indica il tempo di durata del Jitter . Il jitter di clock rappresenta lo sfasamento tra l'istante del telegramma di GC rispetto al tempo atteso calcolato secondo criteri di equidistanza.

T_{DX} (Tempo di Data_Exchange)
Questo tempo rappresenta la somma dei tempi di trasmissione di tutti I telegrammi di Data_Exchange di tutti gli slave.

T_{DP} (Tempo di ciclo DP)
E' la durata del tempo di un ciclo DP

GC: Global_Control
La fine del telegramma di Global Control segna l'inizio del nuovo ciclo DP.

DX: Data_Exchange
Tramite il servizio Data_Exchange, l'utente può scambiare dati tra il master e gli n-slave in sequenza.

MSG: sevizi aciclici
Dopo la trasmissione ciclica il master può trasmettere un servizio aciclico, DPV1.

GAP: Ricerca di ulteriori stazione attive.

TOK: Passaggio del Token
Il token è passato alla propria stazione o ad altri master.

RES: Riservato
Il master trasmette a se stesso. Serve per realizzare un ciclo di bus costante.

3.1.5 SINCRONIZZAZIONE DELLE APPLICAZIONI

Per utilizzare la modalità isocrona è fondamentale tenere in considerazione i tempi di esecuzione delle applicazioni che girano sul master e sugli slave.

Nella figura 3.4 sono indicati i vari tempi necessari alla sincronizzare delle applicazioni. In questo esempio un master DP-V2 controlla tre slave intelligenti che a loro volta comandano tre motori, gli slave tramite degli encoder fornisco il valore attuale della posizione. Sul master girano le applicazioni che controllano la posizione di ogni singolo motore. L'algoritmo di controllo ad anello chiuso nell'applicazione del master è rappresentato da R1-R2-R3. Sullo slave, di tipo intelligente, gira l'applicazione che controlla la velocità. L'applicazione che gira sullo slave 1 è rappresentata da R1, sullo slave 2 da R2 e sullo slave 3 da R3. L'applicazione del master è completamente separata dall'applicazione dello slave.

Al tempo T_i viene acquisito il valore attuale di posizione, mentre al tempo T_o abbiamo la trasmissione del valore di setpoint di posizione all'attuatore.

Vedremo, che impostando opportunamente questi tempi, è possibile ottimizzare i tempi necessari al sistema per il controllo del funzionamento dei motori.

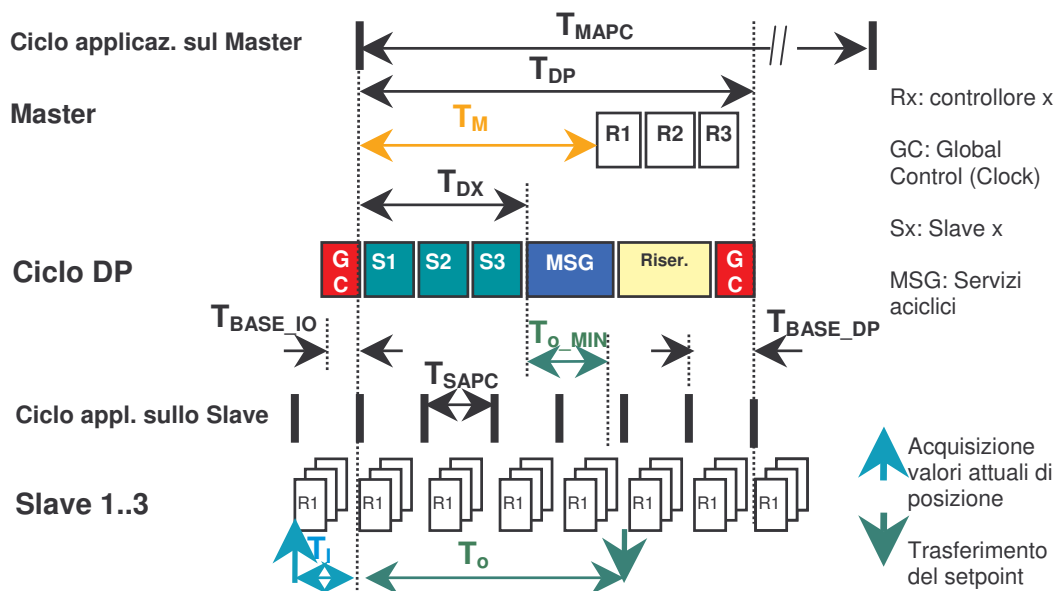



Figura 3.4 – Tempi necessari alla sincronizzazione delle applicazioni

3.1.5.1 DESCRIZIONE DEI TEMPI DA IMPOSTARE IN MODALITÀ ISOCRONA

T_{DP} (Tempo di ciclo DP)

Il tempo di ciclo DP è costituito dalle seguenti parti:

- Durata dei servizi ciclici T_{DX}: dipende dal numero di slave e dalla durata del telegramma DX
- Durata dei servizi aciclici: dipende dalla massima lunghezza dei telegrammi DP-V1
- Durata fino a quando viene generato un nuovo impulso di clock: GAP, passaggio del token, riservato, Global_Control

In aggiunta, deve essere rispettata la condizione seguente:

- $T_{DP} >= maxT_{DP_MIN}$

Dove $maxT_{DP_MIN}$ rappresenta il valore di T_{DP_MIN} più alto fra tutti gli slave.

Impostazione dei tempi nel file GSD.

- T_{BASE_DP} Tempo base di T_{DP} (Unità di misura: [1/12 µs], p.e.: 3000 = 250 µs)
- T_{DP_MIN} Durata minima ciclo DP (Unità di misura: [T_{BASE_DP}], p.e.: 2 = 500 µs)

I valori dei tempi devono soddisfare le condizioni seguenti:

$T_{BASE_DP} = 1\ ms/2^n$ con n = 0 – 5 (numero intero)

Questo significa che il valori ammissibili sono:

T_{BASE_DP} 375/750/1500/3000/6000/12000 =
31.25µs/62.5µs/125µs/250µs/500µs/1000µs

L'azionamento deve essere in grado di trattare i multipli ammissibili del valore selezionato.

T_I (Tempo di Input)

Rappresenta il tempo necessario all'acquisizione del valore degli ingressi di uno slave

Impostazione dei tempi nel file GSD.

- T_{BASE_IO} Tempo base di T_I,T_O (Unità di misura: [1/12 μs], p.e.: 3000 = 250 μs)
- T_{I_MIN} T_I minimo (Unità di misura: [T_{BASE_DP}], p.e.: 1 = 250 μs)

I valori dei tempi devono soddisfare le condizioni seguenti:

$T_{BASE_IO} = 1\ ms/2^n$ con n = 0 – 5 (numero intero)

Questo significa che i valori ammissibili sono:

T_{BASE_IO}: 375/750/1500/3000/6000/12000 =
31.25μs/62.5μs/125μs/250μs/500μs/1000μs

L'azionamento deve essere in grado di trattare i multipli ammissibili del valore selezionato.

T_O (Tempo di Output)

Rappresenta il tempo necessario per trasferire il setpoint sulle uscite di uno slave

Impostazione dei tempi nel file GSD.

- T_{BASE_IO} Tempo base di T_I,T_O (Unità di misura: [1/12 μs], p.e.: 3000 = 250 μs)
- T_{O_MIN} Minimo (T_O-T_{DX}) (Unità di misura: [T_{BASE_DP}], p.e.: 1 = 250 μs)

I valori dei tempi devono soddisfare le condizioni seguenti:

$T_{BASE_IO} = 1\ ms/2^n$ con n = 0 – 5 (numero intero)

Questo significa che i valori ammissibili sono:

T_{BASE_IO}: 375/750/1500/3000/6000/12000 =
31.25μs/62.5μs/125μs/250μs/500μs/1000μs

L'azionamento deve essere in grado di trattare i multipli ammissibili del valore selezionato.

T_M (Tempo di Master)

Rappresenta il tempo dall'inizio del ciclo che il master deve attendere prima di eseguire la sua applicazione.

T_{MAPC} (Tempo di ciclo dell'applicazione sul Master)

Rappresenta il tempo di ciclo dell'applicazione sul master che controlla gli slave.

T_{SAPC} (Tempo di ciclo dell'applicazione sullo Slave)

Rappresenta il tempo di ciclo dell'applicazione sullo slave che legge gli ingressi e scrive le uscite.

Valori differenti di T_{MAPC}, T_{DP}, T_{SAPC}

Nel caso si utilizzino più cicli DP, è possibile specificare dei valori differenti di T_{MAPC}, T_{SAPC}

$T_{MAPC} >= T_{DP} >= T_{SAPC}$

$T_{MAPC} = n * T_{DP}$ (n = 1 - 14)

Tempi di trasmissione

Dipendono dal modello usato, valori pertinenti del tempo di trasmissione possono essere :

- Master -> Slave: 1-2 T_{DP}, dipende da T_O
- Slave -> Master: 1-3 T_{DP}, dipende da T_I e T_M
- Master <-> Slave ritardi aggiuntivi dovuti al passaggio attraverso interfacce HW e SW

Nel caso di funzionalità tecnologiche quali interpolazione di assi è richiesto lo stesso tempo di trasmissione per tutti gli slave.

3.1.5.2 CICLO DP SEMPLICE

Nel ciclo DP semplice si usa il cosiddetto "buffered synchronized isochronous mode". La figura mostra un esempio di controllo di posizione. In questo esempio, sono necessari quattro cicli DP per una reazione nell'anello di controllo della posizione.

- 1° ciclo Acquisizione del valore di posizione (slave)
- 2° ciclo Trasmissione del valore di posizione (slave->master)
- 3° ciclo Elaborazione dell'algoritmo di controllo posizione (master)
- 4° ciclo Trasmissione del setpoint (master->slave) e attuazione del setpoint (slave)

Nel ciclo DP semplice si ha una bassa richiesta di performance computazionali del master. Tuttavia, questo porta ad un incremento del ritardo nel controllo. Il ritardo è calcolato come:

$$Ritardo = 4 * T_{DP}$$

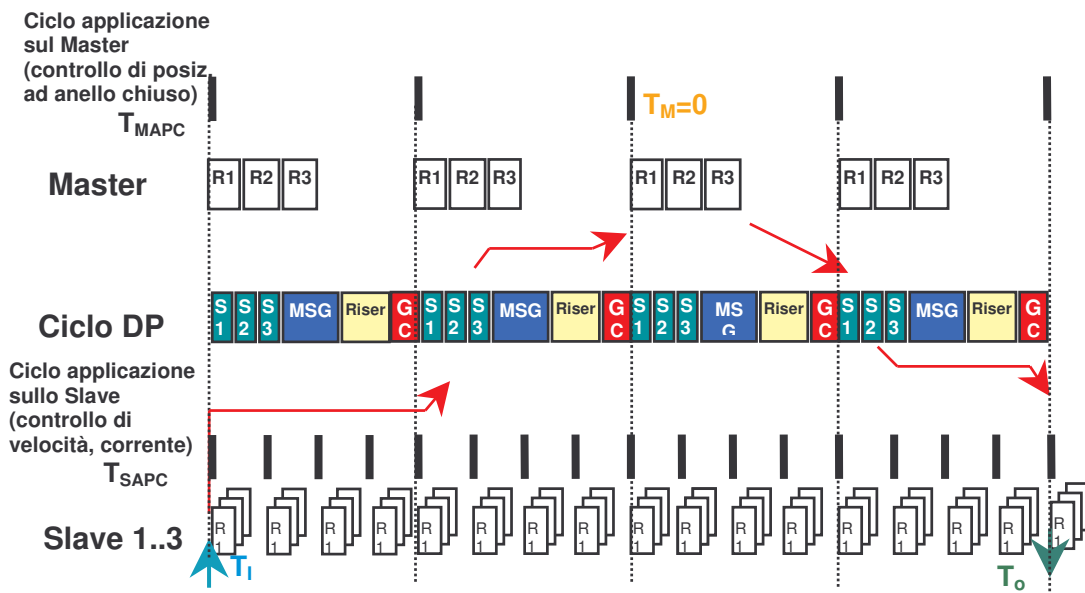



Figura 3.5 – Ciclo DP semplice

3.1.5.3 CICLO DP OTTIMIZZATO

Nel ciclo DP ottimizzato si usa l' "enhanced synchronized isochronous mode".
La sequenza "acquisizione del valore di posizione, trasmissione del valore di posizione, elaborazione dell'algoritmo di controllo, trasmissione del setpoint" è ottimizzata rispetto al tempo al fine di minimizzare il ritardo di controllo. La sincronizzazione del ciclo di clock per applicazioni di classe 4 "Controllo di posizione e interpolazione della traiettoria centralizzato" richiede questa modalità ottimizzata.
Vediamo ora le varie ottimizzazioni:

1. Ottimizzazione del tempo T_I dello slave. Questo tempo, al fine di sincronizzare l'acquisizione dei valori attuali, deve essere posizionato quanto più possibile alla fine del ciclo DP.
2. Ottimizzazione del tempo T_O dello slave. Questo tempo, al fine di sincronizzare l'acquisizione dei setpoint da parte di tutti gli slave, deve essere posizionato quanto più possibile alla fine della trasmissione dei dati ciclici.
3. Ottimizzazione del tempo T_M del master. Il controllo della posizione e' ritardato del tempo T_M, in questo modo i valori di posizione trasmessi sono disponibili all'applicazione di controllo del master all'interno dello stesso ciclo DP.

   Il ritardo può essere realizzato in due modi:

   - la sequenza degli slave all'interno di un ciclo DP è sconosciuta. Il controllo della posizione è ritardato di un tempo sufficiente al completamento della parte ciclica di Data_Exchange per tutti gli slave
   - la sequenza degli slave all'interno di un ciclo DP è nota. Il controllo della posizione può essere effettuato secondo la sequenza di trasmissione dei dati. Il controllo della posizione di un asse può avvenire alla fine del trasferimento dati (Data_Exchange) dello slave corrispondente.

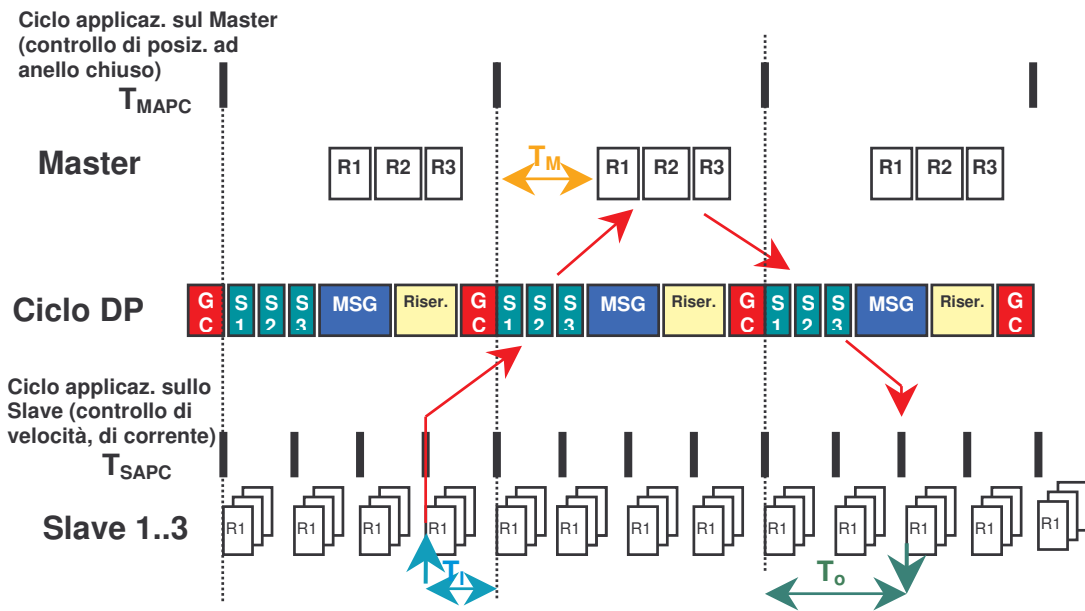



Figura 3.6 Ciclo DP ottimizzato

Questo modello ottimizzato fa grande uso di risorse computazionali, oltre a ciò si richiede un controllo sequenziale del master e degli slave. Come grande vantaggio si ha una minimizzazione del ritardo del controllo. Il ritardo è calcolato come:

$$\text{Ritardo} = T_{DP} + T_I + T_O$$

3.1.5.4 CICLO DP OTTIMIZZATO CON T_{MAPC} = 2 * T_{DP}

In questo esempio il master è sollevato per quanto riguarda le richieste di calcolo, infatti il master ha a disposizione un tempo di calcolo doppio, T_{MAPC} = 2 * T_{DP}. Il ciclo di controllo della posizione sul master è un multiplo del ciclo DP. Questa configurazione può essere necessaria se l'algoritmo di controllo della posizione è solo una piccola parte del programma che gira sul master.

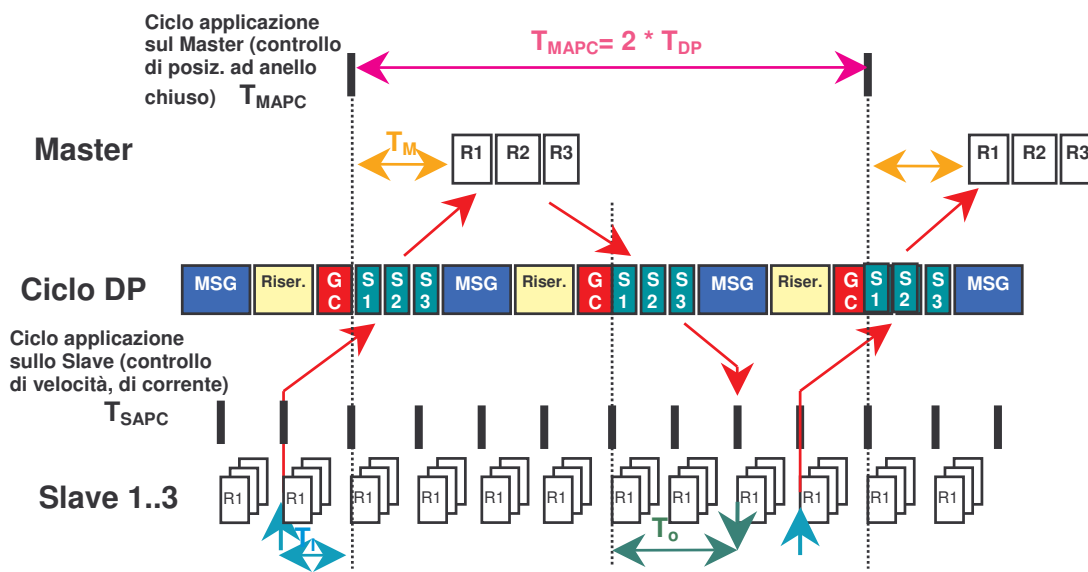



Figura 3.7 – Ciclo DP ottimizzato con T_{MAPC} = 2 * T_{DP}

3.1.6 DATA EXCHANGE BROADCAST (DxB)

La funzionalità di Data Exchange Broadcast permette lo scambio dati diretto fra due o più slave. Questo rende possibile lo scambio dei dati di processo senza passare attraverso il master e la rilocazione degli algoritmi di controllo di anello nello slave.
Nella modalità DxB abbiamo due tipi di slave, il publisher e il subscriber. Il publisher è in genere lo slave che invia i dati di processo ad altre stazioni (subscriber). Il subscriber è uno slave che riceve i dati di processo da uno o più publisher. In un sistema PROFIBUS DP ci possono essere diversi publisher e subscriber. Un publisher può essere allo stesso tempo un subscriber e viceversa.
Un publisher può essere uno slave DP-V0 che supporta il byte 1 di stato del telegramma di parametrizzazione di DP-V1. Un subscriber deve operare come uno slave DP-V1, deve quindi supportare i tre byte di stato del telegramma di parametrizzazione di DP-V1 e l'elaborazione di stato strutturata come per il protocollo DP-V1. Nel paragrafo 3.1.7 verrà descritto in dettaglio il significato dei bit dei byte di stato.

3.1.6.1 COME OPERA IL DxB

La figura mostra l'esempio di un publisher e due subscriber. Uno slave è abilitato come un publisher da un master di classe 1 attraverso il telegramma di parametrizzazione.
Durante lo scambio dati ciclico il master di classe 1 invia i suoi dati allo slave inserendo nell'intestazione del telegramma uno speciale Function Code (FC = x7H).
Il publisher risponde con i suoi dati di processo inserendoli in un telegramma di broadcast con indirizzo di destinazione 127. Questi dati di processo possono essere ricevuti dai subscriber abilitati per quel publisher.
Un subscriber deve essere abilitato come tale durante la fase di start-up, e preleva solamente i dati dal publisher assegnato. In pratica il tipo di struttura che definisce questi parametri è la "Subscribertable" che viene caricata dal master di classe 1 tramite la parametrizzazione strutturata.

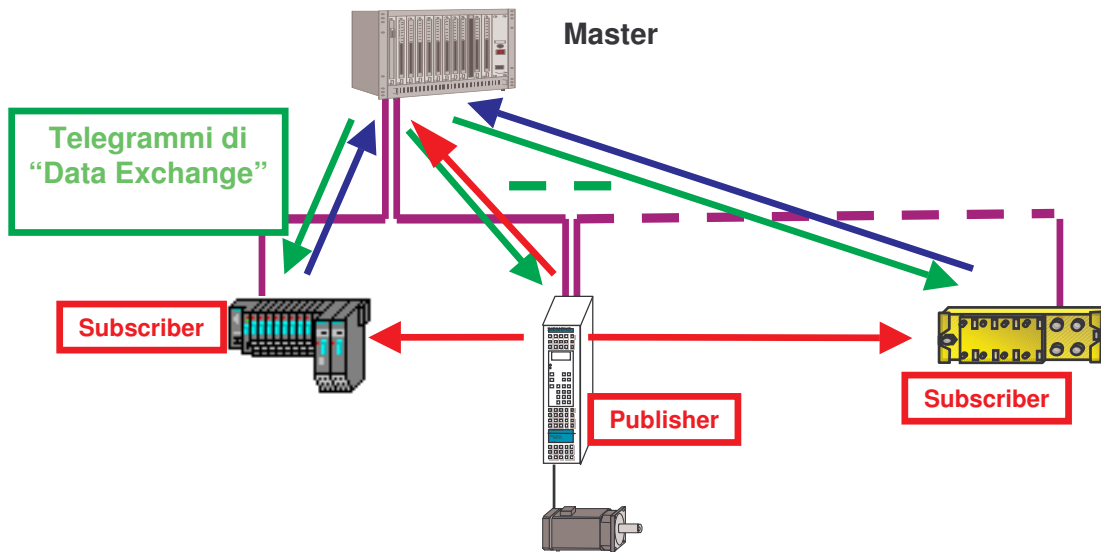
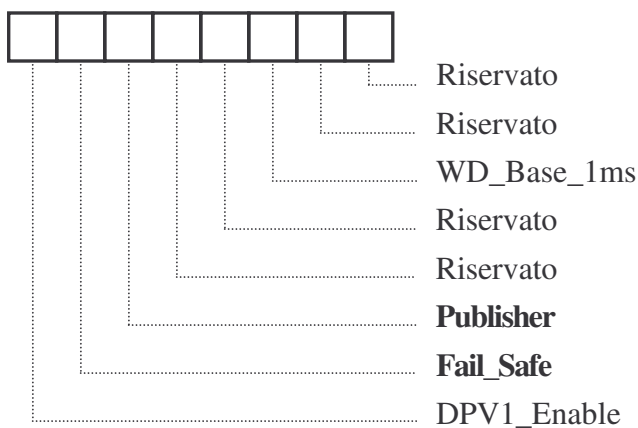



Figura 3.8 – Comunicazione Slave to Slave

3.1.7 PARAMETRIZZAZIONE STRUTTURATA

Un dispositivo di bus di campo che supporta una o più funzionalità del protocollo PROFIBUS DP-V2 deve supportare la parametrizzazione strutturata nella fase di start-up. Nella parametrizzazione strutturata i primi tre parametri utente sono fissi.
I parametri rilevanti per la comunicazione DP-V2 sono mostrati nelle figure seguenti in grassetto. I byte 8..10 del telegramma di parametrizzazione sono i tre byte di stato di DP-V1.

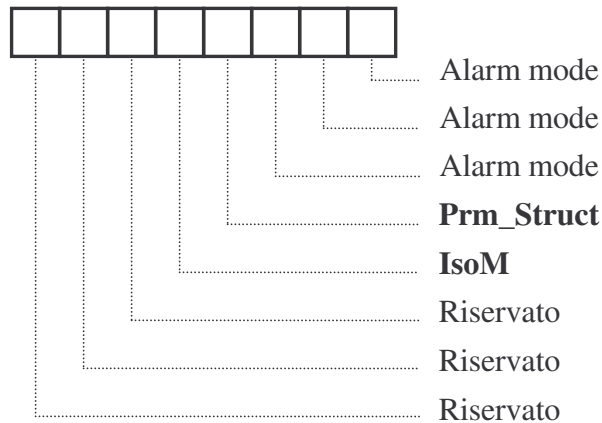
Byte 8 – Primo byte di stato di DP-V1

Riservato
Riservato
WD_Base_1ms
Riservato
Riservato
Publisher
Fail_Safe
DPV1_Enable

Publisher = 1 abilita lo slave al funzionamento come publisher.

Fail_Safe = 1 indica che il master supporta il modo Fail_Safe . Questo è importante per riconoscere che il master non è ancora in modo Operate. Un cambio di stato interno è indicato dal master attraverso un telegramma di broadcast di Global Control.

Byte 9 – Secondo byte di stato di DP-V1

Il byte 9 (Secondo byte di stato di DP-V1) è utilizzato per funzionalità di tipo DP-V1.

Byte 10 – Terzo byte di stato di DP-V1

Prm_Struct = 1 indica che attaccato al telegramma di parametrizzazione c'è la struttura di parametrizzazione

IsoM = 1 indica che lo slave deve operare in modalità isocrona

I parametri strutturati, che seguono i tre byte di stato, sono descritti dalla tabella seguente.

NOME STRUTTURA	COMPOSIZIONE STRUTTURA
Strutured_Prm_Data	Structure_Length, Structure_Type, Slot_Number, Reserved (1 byte), User_Prm_Data
DXB LinkTable	Structured Length, Structure_Type(=3), Slot_Number(=0), Reserved (1 byte), Version(=1), (Publisher_Addr, Publisher_Length, Sample_Offset, Sample_Length)
DXB-Subscribertable:	Structured Length, Structure_Type(=7), Slot_Number(=0), Reserved (1 octet), Version(=1), (Publisher_Addr, Publisher_Length, Sample_Offset, Dest_Slot_Number, Offset_Data_Area, Sam-ple_Length)
Parametri IsoM	**Structured Length, Structure_Type(=4), Slot_Number(=0), Reserved (1 octet), Version(=1),** T_{BASE_DP}, T_{DP}, T_{MAPC}, T_{BASE_IO}, T_I, T_O, T_{DX}, T_{PLL_W}, T_{PLL_D}
PrmCmd	Structured Length, Structure_Type(=2), Slot_Number(=0), Specifier, Function, Properties, Output_Hold_Time

Esempio di struttura (Parametri IsoM) nella modalità isocrona

Byte 1 .. 7	Parametri Standard
Byte 8 .. 10	Bytes di stato DP-V1 1 .. 3
Byte 11	Lunghezza della Struttura = 28 / 0x1C
Byte 12	Tipo di Struttura = 4 (vedi tabella precedente)
Byte 13	Slot = 0
Byte 14	Riservato = 0
Byte 15	Versione = 1
Byte 16.. 19	T_{BASE_DP} = Unsigned 32 (375, 750, 1500, 3000, 6000, 12000 con default1500)[$^1/_{12}$us]
Byte 20/21	T_{DP} = Unsigned 16 (1 to 2^{16}-1) [T_{BASE_DP}]
Byte 22	T_{MAPC} = Unsigned 8 (1 to 14) [T_Dp]
Byte 23.. 26	T_{BASE_IO} = Unsigned 32 (375, 750, 1500, 3000, 6000, 12000 con default 1500)
Byte 27/28	Ti = Unsigned 16 (0;1 to 2^{16}-1| [T_{BASE_IO}]
Byte 29/30	To = Unsigned 16 (0;1 to 2^{16}-1) [T_{BASE_IO}]
Byte 31.. 34	T_{DX} = Unsigned 32 (0;1 to 2^{32}-1) [$^1/_{12}$us]
Byte 35/36	T_{PLL_W} = Unsigned 16 (1 to 2^{16}-1, con default 12) [$^1/_{12}$us]
Byte 37/38	T_{PLL_D} = Unsigned 16 (0 to 2^{16}-1) [$^1/_{12}$us]

3.1.8 IL TELEGRAMMA Ext_User_Prm

Utilizzando la modalità isocrona, la modalità Data Exchange Broadcast, i profili o i parametri utente si raggiunge un elevato numero di parametri che non possono essere trasmessi in un solo telegramma di parametrizzazione. Quindi è stato necessario definire un secondo telegramma di parametrizzazione per spedire tutti i parametri richiesti allo slave durante la sequenza di avvio. Il telegramma Ext_User_Prm ha la stessa struttura del telegramma Set_Prm, la differenza consiste solo in un differente DSAP (0x35). Il telegramma di Ext_User_Prm deve essere spedito fra il telegramma Set_Prm e Chk_Cfg. Se nel file GSD è specificato la linea X_Prm_SAP_supp = 1, allora lo slave deve rispondere ad una struttura valida con 0xE5.

3.1.9 AFFIDABILITÀ DEI DATI

Per garantire l'affidabilità della trasmissione in entrambe le direzioni (master<->slave) si utilizzano diversi metodi e in particolare meccanismi di monitor sul tempo di jitter e sulla durata del ciclo DP.
Per quanto riguarda la possibilita' di riconoscere se un telegramma è arrivato correttamente e nella sequenza giusta, si utilizza un contatore a 4-bit per il master chiamato "Master's Sign-of-Life" (M-LS) e per lo slave un contatore sempre a 4-bit chiamato "Slave's Sign-of-Life".
Il campo dei valori del contatore va da 1 a 15 (0 = non valido).

3.1.9.1 MASTER'S SIGN-OF-LIFE (M-LS)

Questo contatore è incrementato dal master ad ogni ciclo dell'applicazione di controllo della posizione.
Trasmissione del M-LS
Lo slave riceve il nuovo M-LS dal master insieme al nuovo valore di setpoint al tempo TO del ciclo successivo.
Monitoraggio del M-LS

Se l'applicazione che gira sullo slave riconosce un conteggio sbagliato (per esempio riconosce uno scostamento positivo o negativo), viene utilizzata una strategia specifica di gestione dell'errore che dipende dal tipo di dispositivo.
Prima di iniziare la strategia di errore lo slave aspetta per un tempo

$T_{MLS}= n*T_{DP}$

Se durante questo tempo non viene riconosciuto il valore numerico aspettato, viene incrementato un contatore di guasto, e al raggiungimento di un determinato valore di soglia lo slave invia un messaggio di guasto.
Dopo aver riconosciuto il guasto, l'applicazione dello slave tenta di risincronizzarsi automaticamente con il segnale di M-LS dell'applicazione del master.
A seconda dei casi, potrebbe essere necessario un nuovo avvio del sistema.
Il M-LS potrebbe fallire per una delle seguenti ragioni:

- Errore a livello dell'applicazione del master (con la trasmissione DP ancora funzionante)
- Il tempo di ciclo TDP è stato superato (durante la ripetizione di telegrammi)

3.1.9.2 SLAVE'S SIGN-OF-LIFE (S-LS)

Questo contatore è incrementato dallo slave ad ogni ciclo DP.
Trasmissione del S-LS
Il master riceve il nuovo S-LS dallo slave insieme ai dati trasmessi.
Monitoraggio del S-LS
Se l'applicazione che gira sul master riconosce un conteggio sbagliato (p.e. riconosce uno scostamento positivo o negativo), viene utilizzata una strategia specifica di gestione dell'errore che dipende dal tipo di dispositivo.
Prima di iniziare la strategia di errore lo slave aspetta per un tempo

$T_{SLS}= n*T_{DP}$

Generalmente accade, come nel caso precedente, che ad ogni errore viene incrementato un contatore di guasto, e al raggiungimento di un determinato valore di

soglia il master spegne l'azionamento che ha segnalato l'errore e manda una appropriata segnalazione di guasto all'utente.

Dopo aver riconosciuto il guasto, l'applicazione del master tenta di risincronizzarsi automaticamente con il segnale di S-LS dell'applicazione dello slave.

A seconda dei casi, potrebbe essere necessario un nuovo avvio del sistema.

Il S-LS potrebbe fallire per una delle seguenti ragioni:

- Errore a livello dell'applicazione dello slave (con la trasmissione DP ancora funzionante)
- Errore di tipo DP (lo slave non risponde nonostante la ripetizione di telegrammi)

3.1.9.3 STRATEGIA DI CONTEGGIO DEI CONTATORI SIGN-OF-LIFE (LS)

Per evitare degli spegnimenti dovuti ad errori sporadici si utilizza una strategia di conteggio. Questa strategia garantisce che deve esserci almeno una percentuale specifica di telegrammi sbagliati prima di arrestare l'azionamento.

Il contatore di guasto è incrementato ad ogni errore del Sign-of-Life (scostamento positivo o negativo) di 10. Se nel ciclo successivo il LS è corretto, il contatore è decrementato di 1. Il contatore può essere decrementato fino a 0, valore di inizio del contatore. **Questa strategia assicura che più del 90% dei telegrammi sono stati trasferiti in maniera corretta**.

Il massimo numero di errori di LS consecutivi che possono essere accettati prima di segnare un guasto o il valore massimo del contatore di guasto possono essere decisi in fase di parametrizzazione. Il numero di errori consecutivi è funzione anche della storia precedente, in alcuni casi sono sufficienti pochi errori di LS (vedi figura) per far arrestare l'azionamento. Nell'esempio seguente (Fig. 3.9), il contatore di guasto di un azionamento è analizzato in funzione del tempo. Il valore massimo del contatore è stato impostato in fase di parametrizzazione al valore 31.

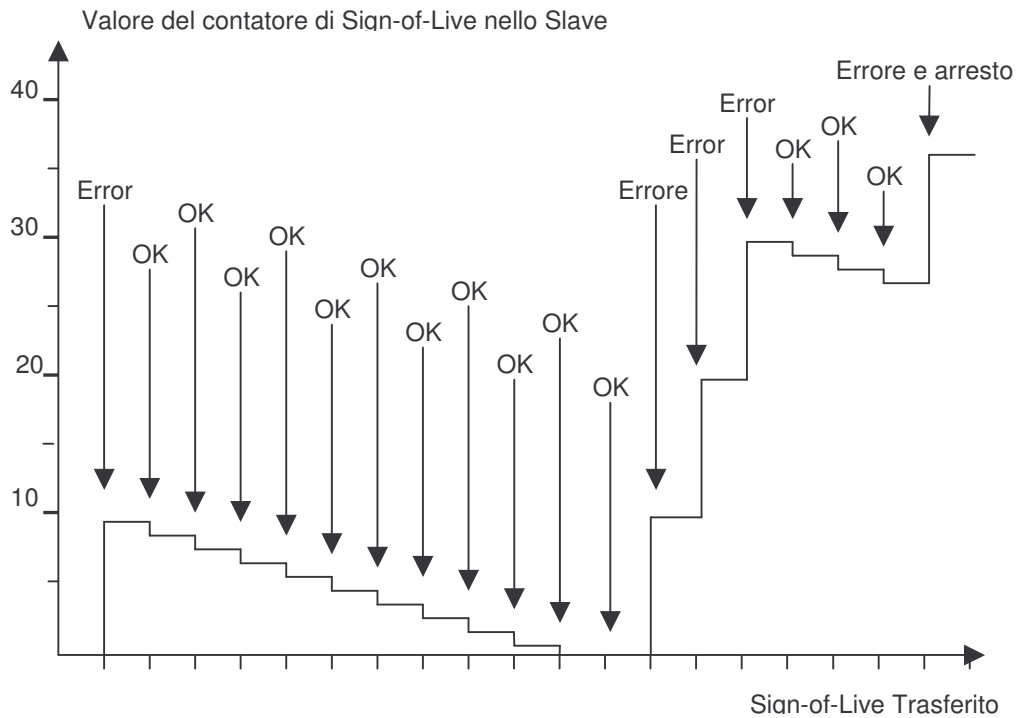



Figura 3.9 – Esempio di funzionamento del contatore Sign-of.live

3.1.9.4 CONTROLLO DEL JITTER IN MODALITÀ ISOCRONA

Attraverso un PLL, la durata del jitter può essere attenuata e sia sporadici errori di clock quanto effetti di runtime possono essere compensati.

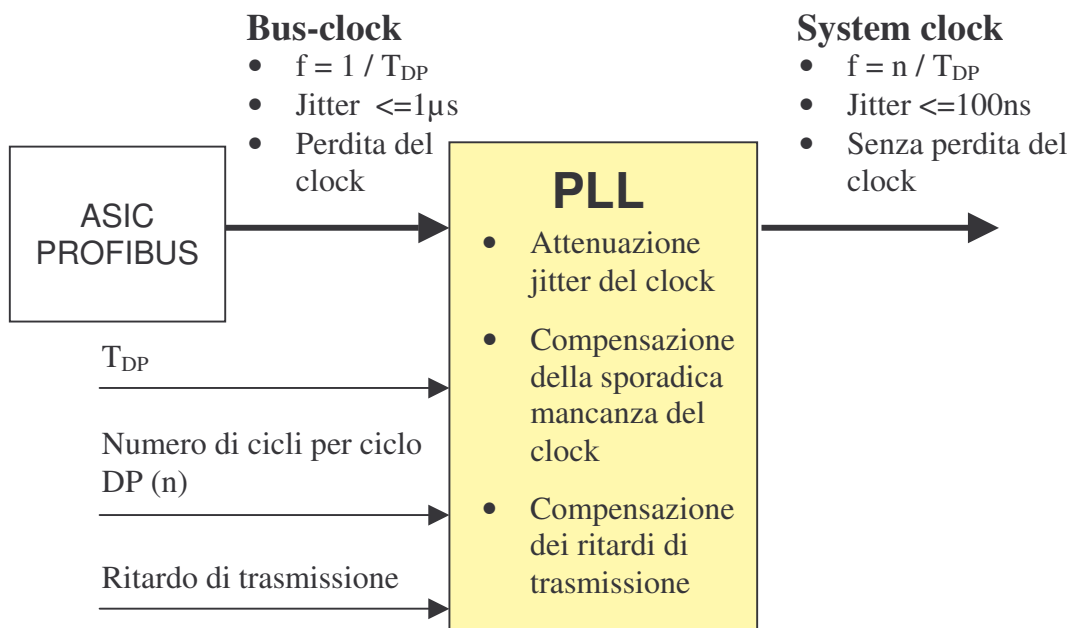



Figura 3.10 – PLL

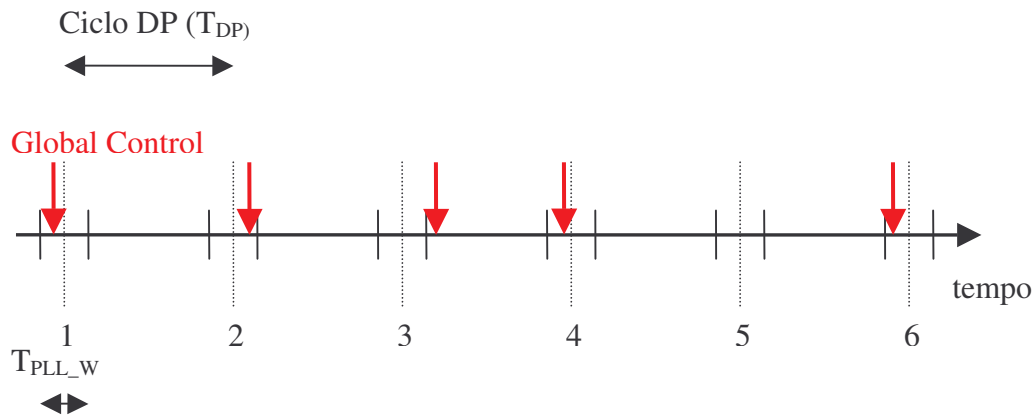



Figura 3.11 – Esempio di errore del telegramma di sincronizzazione

Durante lo start-up del sistema il parametro T_{PLL_W} nel telegramma di parametrizzazione del master di classe 1 (DPM1) indica l'ampiezza della finestra temporale, entro la quale il jitter si può muovere. Nella figura 3.11 vengono mostrati 6 cicli DP. Il ciclo 3 mostra un errore di ritardo. Il segnale di clock arriva al di fuori della finestra di clock configurata. Il ciclo 5 mostra la mancanza del segnale di clock. Possibili reazioni possono essere un retriggheraggio del PLL, il settaggio del contatore LS a 0, o il passaggio allo stato safe.

3.1.9.5 VIOLAZIONE DELLA DURATA DEL CICLO DP

La violazione della durata del tempo di ciclo T_{DP} (per esempio a causa di ripetizioni di telegrammi) può in alcuni casi essere gestita.

Come mostrato in figura, una violazione della durata del ciclo DP può non condurre ad una deviazione del ciclo di clock.

Il master deve assicurare che non avvengano deviazioni permanenti del ciclo di clock a causa di una violazione delle sua durata. Il ciclo deve essere "rimesso al suo posto" dal ciclo successivo sopprimendo la parte aciclica posta dopo la violazione.

DP Clock (sincr.)	c / ac	c / ac	c / ac	c / ac	c / ac	c / ac
DP Clock (reale)	c / ac	c / ac	c	c / ac	c / ac	c / ac

c / ac = ciclica / aciclica

3.2 CALCOLO DELLE PRESTAZIONI DI UN SISTEMA PROFIBUS DP-V2

Per valutare le prestazioni di un sistema PROFIBUS DP-V2 è stata analizzata un'applicazione in un impianto di produzione, realizzata tramite una rete di comunicazione con topologia lineare dotata di un Master e 31 Slave. Valuteremo quindi il tempo di aggiornamento del sistema (tempo necessario alla trasmissione delle uscite e all'acquisizione degli ingressi del sistema). Il tempo calcolato fa riferimento solo allo scambio di dati real-time isocroni, per calcolare il tempo totale del ciclo isocrono dovranno essere sommati i tempi di comunicazione riservati allo scambio di dati non isocrono (comunicazione asincrona, Token ecc.), che sono impostabili in fase di configurazione del sistema.
Prima di iniziare l'analisi viene presentata una lista dei tempi di bus che devono essere rispettati per PROFIBUS DP. L'utente non ha influenza su di essi, i tempi in oggetto vengono implementati in fase di progetto del dispositivo.

T_{bit} = tempo necessario per trasmettere un bit: 1/bit rate

T_{SYN} = tempo di sincronizzazione; periodo di sincronizzazione dovuto all'invio di una serie di 33 bit (idle bit)

T_{id1} = tempo di attesa; dopo aver spedito l'ultimo carattere di un messaggio bisogna aspettare questo tempo prima che sia possibile spedire il prossimo messaggio. Il tempo deve essere almeno uguale al tempo T_{SYN} piu un margine di sicurezza, che è specialmente importante nel caso di alta velocità di trasferimento
T_{SDR} = tempo richiesto dallo slave per rispondere. Deve essere compreso fra i valori definiti dallo standard min_{TSDR} e max_{TSDR}

min_{TSDR} = tempo di attesa minimo che uno slave deve far trascorrere prima gli sia permesso di rispondere.

max_{TSDR} = tempo massimo entro il quale uno slave deve rispondere. Per ragioni di ottimizzazione il file GSD deve specificare il valore di questo tempo accettato dal dispositivo. Questo tempo influenza il tempo di risposta e la qualità di un dispositivo di campo e dipende essenzialmente dall'ASIC utilizzato. Per la comunicazione con DP-V2 è assolutamente necessario che il valore inserito sia l'ottimo.

La tabella mostra i valori definiti nello standard PROFIBUS espressi nell'unità di misura $T_{bit.}$

Velocità di trasmissione (Kbit/s)	≤ 187,5	500	1500	3000	6000	12000
min $_{TSDR}$ (T_{bit})	11	11	11	11	11	11
max $_{TSDR}$ (T_{bit})	60	100	150	250	450	800

3.2.1 TEMPISTICHE DI UN SISTEMA PROFIBUS

Il tempo di aggiornamento dipende essenzialmente dai seguenti fattori:

- Velocità di trasmissione dei dati
- Tipo di trasmissione (Half Duplex)
- Tempo di propagazione
- Lunghezza dei dati di Output
- Lunghezza dei dati di Input
- max_{TSDR}
- Min_Slave_Intervall

Per essere sicuri che lo slave possa rispondere all'interno di un ciclo di dati viene impostato il parametro Min_Slave_Intervall. Al giorno d'oggi, utilizzando un Asic è possibile raggiungere valori di Min_Slave_Intervall di circa 100 µs.

Ciascun telegramma consiste in un certo numero di caratteri, ognuno dei quali caratterizzato da 11 bit:

1°: bit di start (ST) che assume sempre valore binario “0”

2°…9°: individuano gli 8 bit di informazione (I)

10°: bit di parità (P)

11°: bit di stop (SP) che assume sempre valore binario “1”

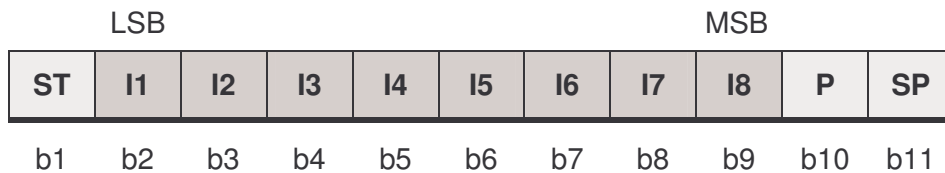



Fig. 2.9 - Frame Carattere

Per il calcolo del tempo di risposta consideriamo che lo scambio dati avvenga tramite telegrammi con campo dati di lunghezza variabile che hanno la seguente struttura:

Telegramma di richiesta

SD2	LE	LEr	SD2	DA	SA	FC	DATA_UNIT	FCS	ED

Telegramma di risposta

SYN: periodo di sincronizzazione (33 bits (*idle bits*) ciascuno di valore “1”).

SD2: (Start Delimiter) delimitatore iniziale, di valore esadecimale pari a 68H.

LE: numero di UC da DA incluso fino a FCS escluso; valori permessi [4…249].

LEr: numero di UC da DA incluso fino a FCS escluso ripetuto

DA: (Destination Address) indirizzo del destinatario [0…127].

SA: (Source Address) indirizzo del mittente [0…126].

FC: (Frame Control) frame di controllo.

DATA_UNIT: campo dati caratterizzato da un numero massimo di caratteri pari a 246.

FCS: (Frame Check Sequence) frame di controllo della sequenza dei bit.

ED: (End Delimiter) delimitatore finale, di valore esadecimale pari a 16H.

Per calcolo dei tempi di risposta ho considerato i seguenti parametri:

$$N_{bit-richiesta_n} = N_{bit-SYN} + N_{bit-HEADER} + N_{byte-richiesta_n} \cdot 11 + N_{bit-TRAILER}$$

$$N_{bit-risposta_n} = N_{bit-HEADER} + N_{byte-rissposta_n} \cdot 11 + N_{bit-TRAILER}$$

$N_{bit-HEADER}$ = 77 (SD2 – LE – LEr – SD2 – DA – SA – FC)

$N_{bit-TRAILER}$ = 22 (FCS – ED)

N_s = numero totale degli slave

V_{Tra} = velocità di trasmissione

L = lunghezza del segmento

V_{Prop} = velocità di propagazione minima pari a 200000000 m/s

Per cui:

$$T_{AGG_PROFIBUS} = \sum_{n=1}^{N_S} \left(N_{bit-richiesta_n} \frac{1}{V_{tra}} + \max_{TSDR\,n} + N_{bit-risposta_n} \frac{1}{V_{tra}} + T_{id1_n} \right) + \frac{L}{V_{Prop}}$$

Capitolo 4

PROFINET I/O

Introduzione

PROFINET è uno standard innovativo e aperto sviluppato da PROFIBUS International per l'automazione industriale basato su Industrial Ethernet.
Diventato standard internazionale con la normativa IEC 61158, supporta attraverso la comunicazione Ethernet l'integrazione di dispositivi di campo per automazione e applicazioni critiche dal punto delle temporizzazioni.
PROFINET è composto da due differenti protocolli : un protocollo **PROFINET CBA** (descritto in appendice C) poco adatto al tempo reale ma basato su standard di accesso molto diffusi quali RPC/DCOM/OPC, e un protocollo pensato per il campo e il tempo reale **PROFINET IO**.

4.1 COMUNICAZIONE IN PROFINET

La comunicazione in PROFINET presenta tre livelli di prestazioni:

1. TCP, UDP e IP per dati non critici rispetto al tempo, come assegnamento di parametri e configurazione
2. Soft Real Time (SRT) per dati di processo critici rispetto al tempo; utilizzata nel campo dell'automazione aziendale
3. Isochronous Real Time (IRT) per particolari applicazioni, come per applicazioni di Motion Control.

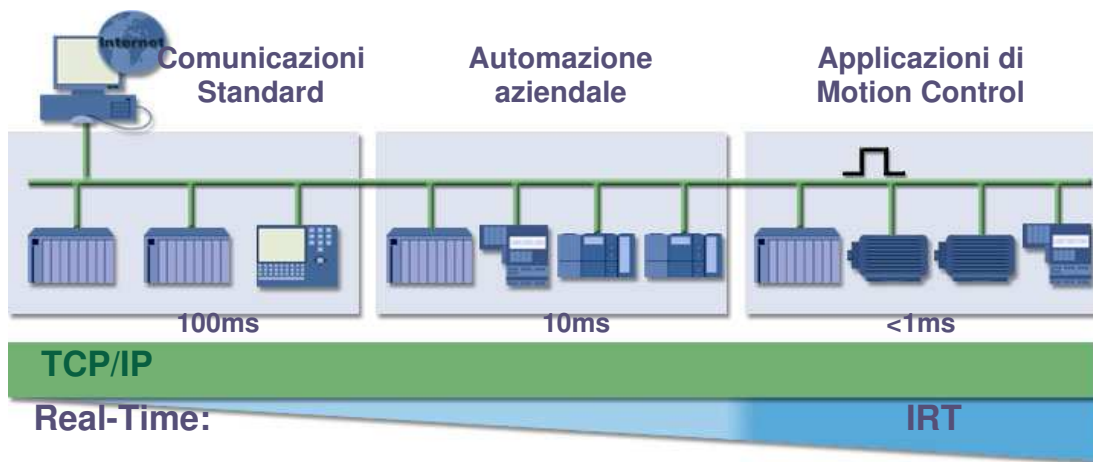



Figura 4.1 – I tre livelli della comunicazione PROFINET

Il grafico sottostante (fig. 4.2) evidenzia la distribuzione dei tempi di risposta dei tre diversi approcci PROFINET alla comunicazione.

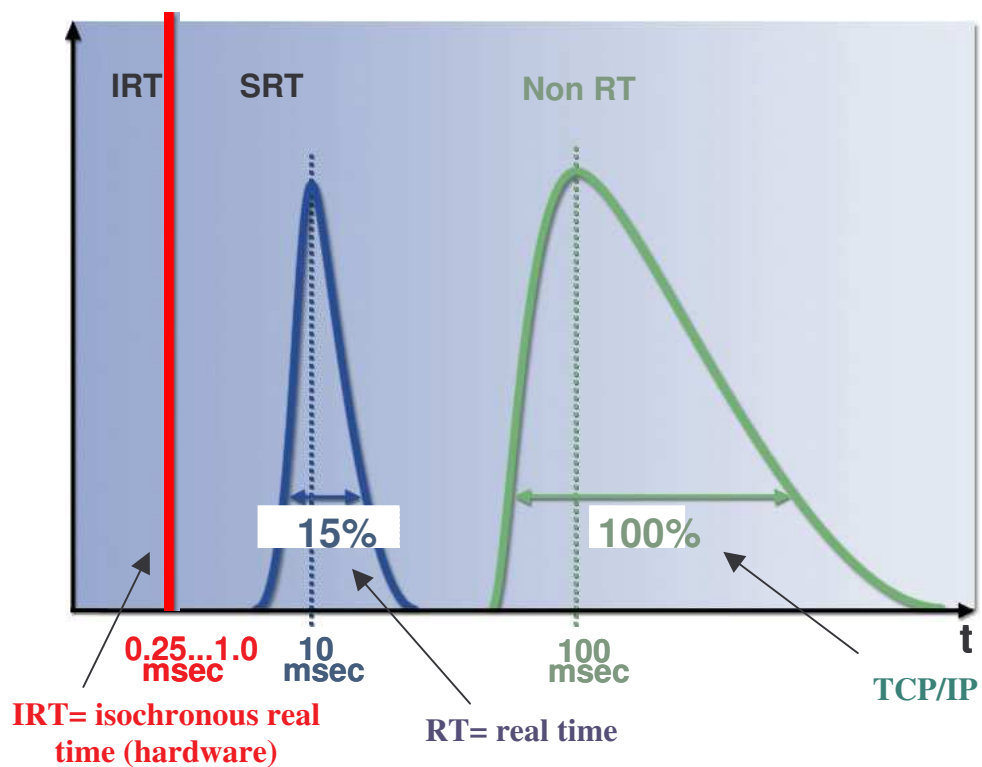



Figura 4.2 - Distribuzione dei tempi di risposta dei tre approcci PROFINET alla comunicazione

La comunicazione standard attraverso il protocollo TCP\IP da tempi di risposta medi attorno ai 100 ms.
La comunicazione tramite SRT Channel evidenzia un miglioramento dei tempi di trasmissione di un fattore 10, i tempi di risposta si attestano infatti attorno ai 10 ms.
Anche la varianza subisce una riduzione che si attesta attorno ad un fattore che oscilla tra le 5 e le 8 volte quello della comunicazione standard.
Pertanto i tre livelli di prestazioni di PROFINET coprono l'intero campo delle applicazioni per l'automazione e di conseguenza le caratteristiche fondamentali dello standard PROFINET sono le seguenti:

- Uso in contemporanea di comunicazioni basate su TCP e comunicazioni real time
- Protocollo di comunicazione real time standardizzato per tutte le applicazioni, sia per comunicazioni tra componenti in sistemi distribuiti, sia per comunicazioni tra i controllori e periferiche decentrate
- Comunicazione real time scalabile

In particolare la caratteristica di scalabilità e di standardizzazione della comunicazione sono uno dei punti di forza di PROFINET. Permettono l'utilizzo del protocollo per tutti i livelli della piramide CIM (§ 1.1.2), dal livello 4, dove la mole di dati scambiata è rilevante, al livello 1, dove sono rilevanti i tempi di risposta, ad esempio per il controllo di un azionamento in un processo automatizzato.

4.1.1 LA COMUNICAZIONE IN TEMPO REALE

PROFINET usa Ethernet e TCP/IP come base per la comunicazione.
Con Ethernet, non è possibile definire un tempo all'interno del quale i dati devono essere trasmessi attendibilmente. L'utilizzo del protocollo TCP/IP ha rivelato che sono necessari tempi considerevoli utilizzando questo stack di comunicazione, oltre a questo l'elaborazione dello stack TCP/IP produce un ritardo non costante. L'esperienza indica inoltre che la velocità della trasmissione seguendo la linea in Ethernet di 100 Mbit è trascurabile rispetto all'elaborazione nei dispositivi.

Ciò significa che tutti i miglioramenti nel tasso d'aggiornamento e quindi nella risposta in tempo reale devono essere realizzati soprattutto attraverso ottimizzazioni dello stack di comunicazione nel Provider e nel Consumer.

Si definisce **tempo di aggiornamento** il tempo che passa da quando una variabile è generata in un'applicazione del dispositivo a quando è trasmessa ad un altro dispositivo attraverso i sistemi di comunicazione e successivamente è messa a disposizione dell'applicazione (si veda la figura 4.3).

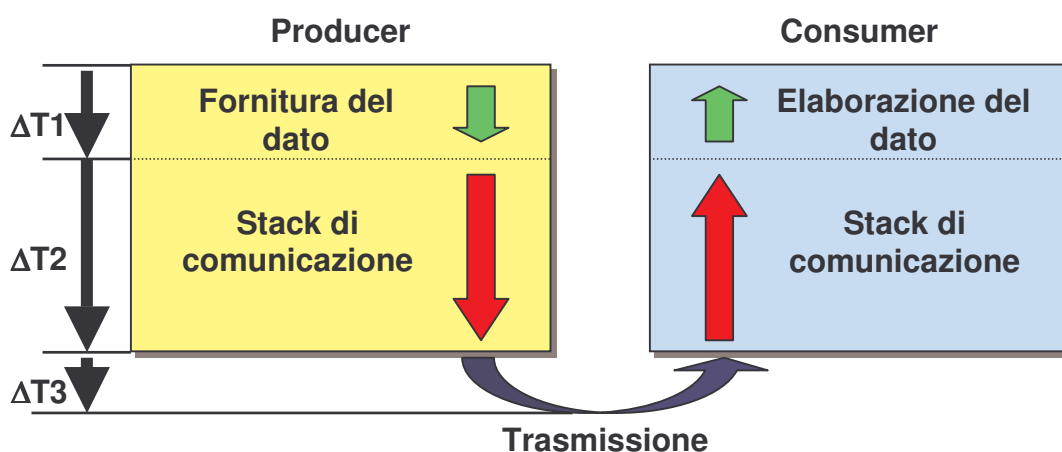



Figura 4.3 – Tempo di aggiornamento

Il tempo ΔT1 è determinato dall'architettura hardware del dispositivo e difficilmente dipende dal protocollo. Il tempo ΔT3 dipende dal sistema di trasmissione. Il tempo ΔT2 è dovuto allo stack di comunicazione.

In PROFINET i dispositivi con funzionalità altamente tempo-critiche possono, quando stabiliscono il collegamento, negoziare protocolli di comunicazione con capacità real-time che riducono il tempo ΔT2.

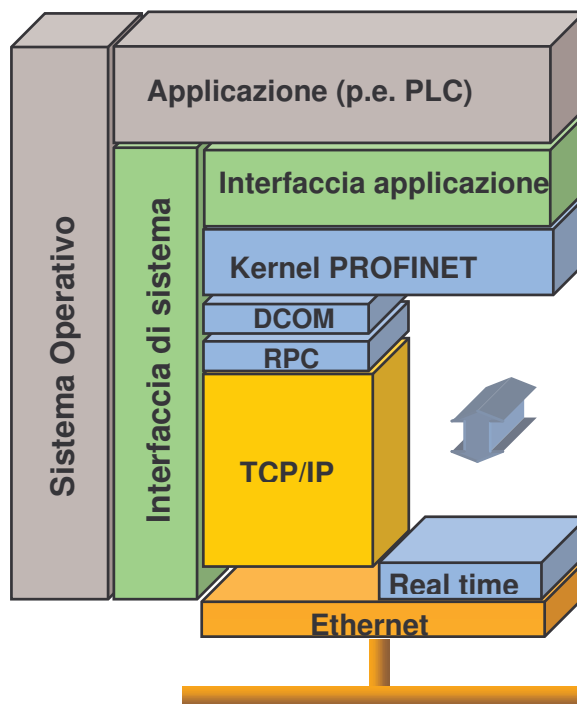



Figura 4.4 - stack di comunicazione PROFINET

In questo modo, PROFINET soddisfa le elevate richieste real-time disposte sul sistema di comunicazione. La soluzione proposta minimizza notevolmente i tempi dello stack di comunicazione e i risultati nelle prestazioni aumentano in termini di tasso d'aggiornamento dei dati di automazione.

L'eliminazione di parecchi livelli di protocollo riduce la lunghezza di messaggio e meno tempo è richiesto prima che i dati da trasmettere siano pronti per la trasmissione e l'applicazione sia pronta per procedere. Un'ulteriore beneficio si ha considerando che la potenza del processore necessario nel dispositivo per la comunicazione è notevolmente ridotto.

PROFINET distingue diverse classi realtime che si differenziano in funzione delle performance richieste: *RT (chiamato anche SRT) e IRT*.

Viene utilizzata anche la seguente terminologia: *Real-time locale* per la comunicazione RT e *Real-time sincronizzato* per la comunicazione IRT

4.2 PROFINET IO

Con PROFINET IO, l'integrazione di dispositivi di campo decentralizzati è implementata direttamente su Ethernet. Per questo scopo, la metodologia di accesso Master-Slave di PROFIBUS DP è stata convertita al modello provider-consumer.
PROFINET prevede tre tipi di dispositivi:

- **IO-Controller:** dispositivo controllore sul quale gira il programma di automazione
- **IO-Device:** dispositivo di bus di campo remoto, che viene assegnato a un IO-Controller
- **IO-Supervisor:** dispositivo/PC di programmazione con funzioni di configurazione e di diagnostica

Dal punto di vista della comunicazione, tutti i dispositivi su Ethernet sono trattati allo stesso modo. Tuttavia, durante il processo di configurazione i vari dispositivi sono assegnati ad un controllore centralizzato (IO-Controller).
Per la configurazione viene utilizzata la stessa interfaccia di PROFIBUS. Durante il funzionamento, i dispositivi periferici (IO-Device) trasferiscono le informazioni di input al controller; il controller elabora le informazioni e trasferisce l'output ai dispositivi periferici.

Le varie informazioni possono essere trasferite tra gli IO-Controller e gli IO-Device attraverso i seguenti canali:

- **Dati ciclici di IO**. Questi dati sono trasferiti sul canale real time
- **Allarmi.** Sono trasferiti sul canale real time
- **Parametrizzazione, configurazione, lettura delle informazioni diagnostiche.** Questi dati sono trasferiti attraverso canali standard sulla base di UDP/IP

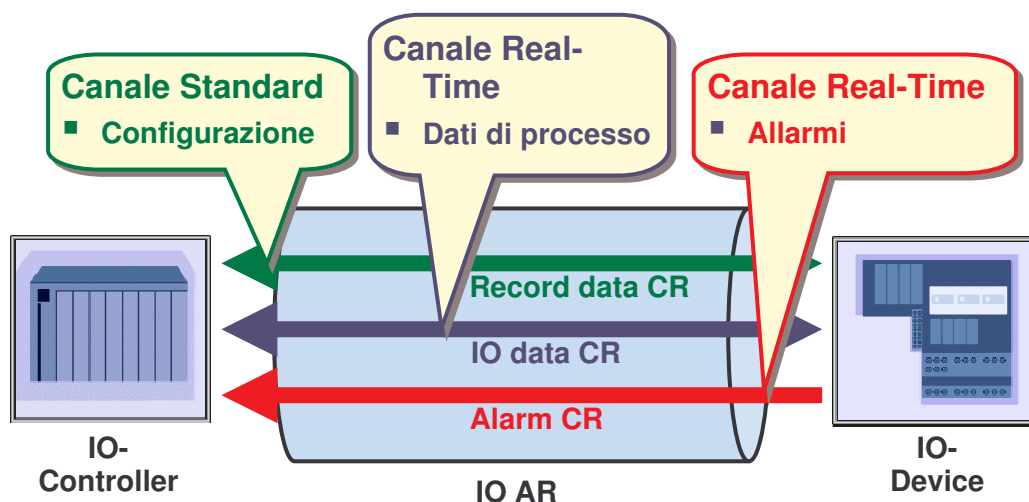



Figura 4.5 – I canali di comunicazione

Alla partenza, vengono stabilite le istanze di relazione (IO-AR) tra gli IO-Controller e gli IO-Device attraverso in canale UDP/IP. L'istanza di relazione IO-AR contiene diverse relazioni di comunicazione (CR) attraverso le quali sono trasferite la configurazione, i dati di processo e gli allarmi. Il dispositivo IO-Controller trasferisce i dati di parametrizzazione e di configurazione del dispositivo IO-Device assegnato sulla "Record data CR". La trasmissione ciclica dei dati di ingresso e di uscita è implementata sulla "IO CR" . Gli eventi aciclici sono trasmessi sulla "Alarm CR" al IO-Controller. In PROFINET esistono diversi tipi di allarme: unplug, plug in, diagnostici, di stato e allarmi di aggiornamento, inoltre sono comunque possibili allarmi specifici inseriti dal produttore del dispositivo. E' possibile assegnare una priorità alta o bassa agli allarmi.

4.2.2 SCAMBIO DI DATI CICLICO

Lo scambio dati ciclico è implementato attraverso la relazione di comunicazione IO CR. I dati di I/O sono trasmessi a cicli di tempo prefissati la cui frequenza di aggiornamento può essere diversificata per ciascun dispositivo. Inoltre l'intervallo di invio può essere diverso dall'intervallo di ricezione.

Per la descrizione di uno scambio di dati ciclico PROFINET IO definisce i seguenti termini, illustrati anche in figura 4.6:

Ciclo di bus: La base tempi del clock è 31.25 μs. Un valore tipico è 32 per avere un ciclo di bus di 1 ms,

Reduction Ratio: Il rapporto di riduzione indica ogni quanti cicli di bus viene spedito un frame. Il ciclo di invio è calcolato moltiplicando il rapporto di riduzione per la durata del ciclo di bus

Frame Offset: Il Frame Offset indica l'offset relativo (in incrementi di 250 ns) dall'inizio del ciclo di bus. Indica quando un determinato frame verrà spedito all'interno del ciclo di bus

Fase: La fase indica in quale ciclo di bus il frame corrispondente verrà spedito. Valori tipici sono 1 o 4.

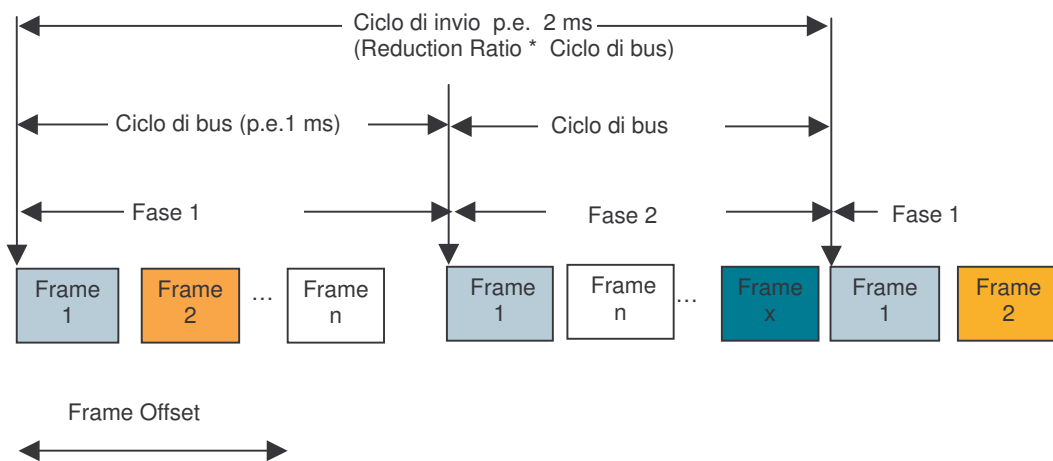



Figura 4.6 – Scambio di dati ciclico

Non tutti i dati devono essere trasmessi con la stessa frequenza, se ciò avvenisse, la stazione più lenta determinerebbe la velocità di spedizione dei dati. Per esempio, i valori di temperatura non devono essere spediti durante ogni ciclo di bus. La soluzione a questo problema è fornita dal "Reduction Ratio" (rapporto di riduzione) che deve essere considerato come un fattore moltiplicatore del ciclo di invio o di ricezione e determina in quale ciclo di quelli successivi verranno trasmessi i dati.

Il rapporto di riduzione è specificato, durante la configurazione, per ogni relazione di comunicazione I/O CR.

4.2.3 MODELLO DEI DISPOSITIVI

Ad ogni dispositivo deve essere assegnato un numero di identificazione, Device Ident (ID) Number, univoco. Questo numero a 32 bit è diviso in due parti: i primi 16-bit rappresentano il numero identificativo del produttore, i rimanenti 16-bit il numero identificativo del dispositivo.
E' stato specificato un modello uniforme per i dispositivi IO-Device PROFINET, che permette una configurazione dei singoli moduli del dispositivo. Questo modello è simile alle caratteristiche del modello PROFIBUS DP. Un dispositivo di campo può essere modulare, cioè includere slot fisici per l'inserzione dei moduli, oppure non espandibile e allora gli slot possono essere considerati virtuali.
In ogni caso questo tipo di modellizzazione permette di indirizzare i canali di I/O presenti nel sistema con indice univoco.

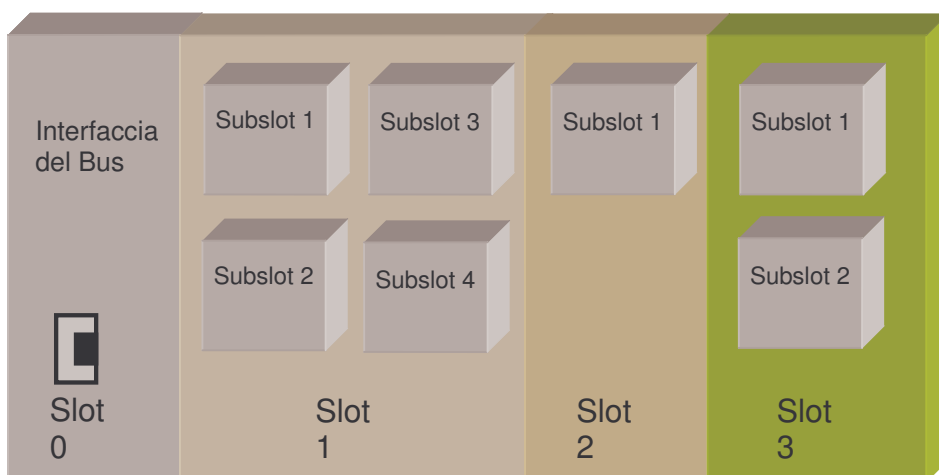



Figura 4.7 – Modello di un IO-Device

Tra l'altro questo progetto modulare fa si che eventuali moduli I/O esistenti su PROFIBUS DP possano essere incorporati in PROFINET senza restrizioni e modifiche. Tutto questo assicura una protezione degli investimenti per i produttori di dispositivi e per gli utenti finali.

Un dispositivo PROFINET è integrato nel tool di configurazione allo stesso modo di un dispositivo PROFIBUS DP tramite una descrizione del dispositivo. Le

caratteristiche del dispositivo sono descritte in un file GSD (General Station Description), che contiene tutte le informazioni necessarie. Il file GSD è in formato XML.

4.2.4 DIAGNOSTICA

PROFINET IO supporta un concetto di diagnostica multi-layer che rende possibile un'efficace metodo di rilevamento degli errori e la conseguente risoluzione di problemi. In figura 4.8 viene illustrato il concetto di diagnostica multi-layer.

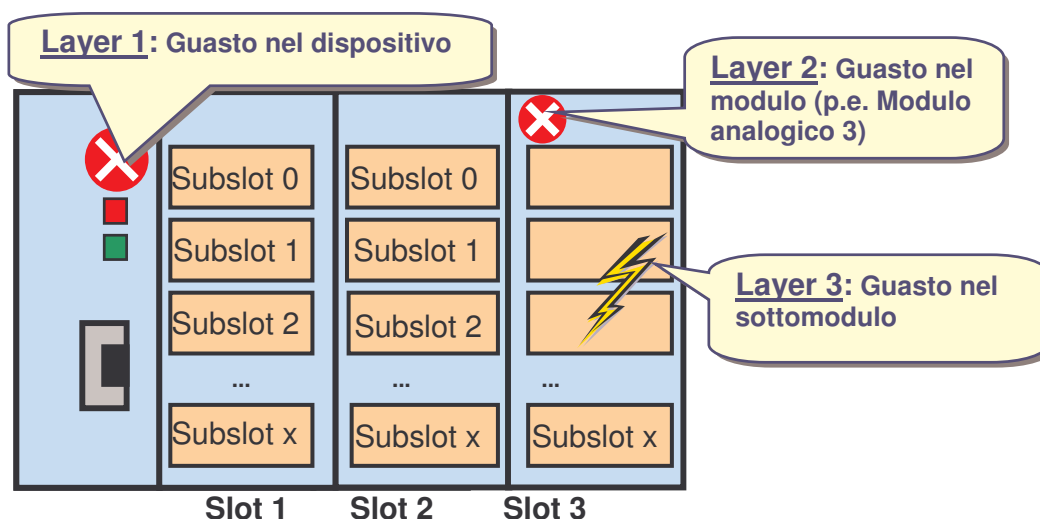



Figura 4.8 - Concetto di diagnostica multi-layer di un oggetto PROFINET IO

Quando si verifica un errore, l'IO Device interessato invia un allarme diagnostico all'IO Controller. Questo allarme richiama una corrispondente routine all'interno del programma del PLC che gestisce il guasto. Dopo aver eseguito la routine l'IO-Controller conferma il riconoscimento dell'errore tramite un telegramma di acknoledgment all IO-Device. Se un difetto in un dispositivo o in un modulo implica la necessità di rimpiazzarlo completamente, l'IO Controller automaticamente effettua la parametrizzazione e la configurazione del nuovo dispositivo o modulo.
Le informazioni diagnostiche sono strutturate in modo gerarchico:

- Numero di slot (modulo)

- Numero di canale
- Tipo di canale
- Numero di codice del guasto
- Informazioni addizionali specifiche del produttore

4.2.5 COMUNICAZIONE RT (SOFT REAL TIME "SRT") IN PROFINET IO

Il cosiddetto Soft Realtime Channel (SRT Channel) è un primo approccio al problema della gestione della comunicazione realtime già implementato nella versione PROFINET V2 e in grado di soddisfare processi nei quali:

- Risultano accettabili ritardi contenuti entro determinati limiti (circa 10 msec)
- Le condizioni imposte sui limiti accettabili per i ritardi debbano essere mantenuti non rigidamente ma attorno a un valore medio.

Si tratta di una soluzione completamente software in grado di soddisfare, per le caratteristiche che presenta, le applicazioni tipiche dell'area della factory automation ove i tempi di risposta del sistema si devono attestare attorno alla decina di millisecondi. In questo caso in PROFINET c'è soltanto uno stack di comunicazione minimizzato e ottimizzato che sostituisci i livelli TCP/IP e UDP/IP.

4.2.5.1 STRUTTURA DEL PACCHETTO REAL-TIME (RT)

PROFINET usa pacchetti Ethernet (§ 2.3.3.2) che sono identificati attraverso l'EtherType.

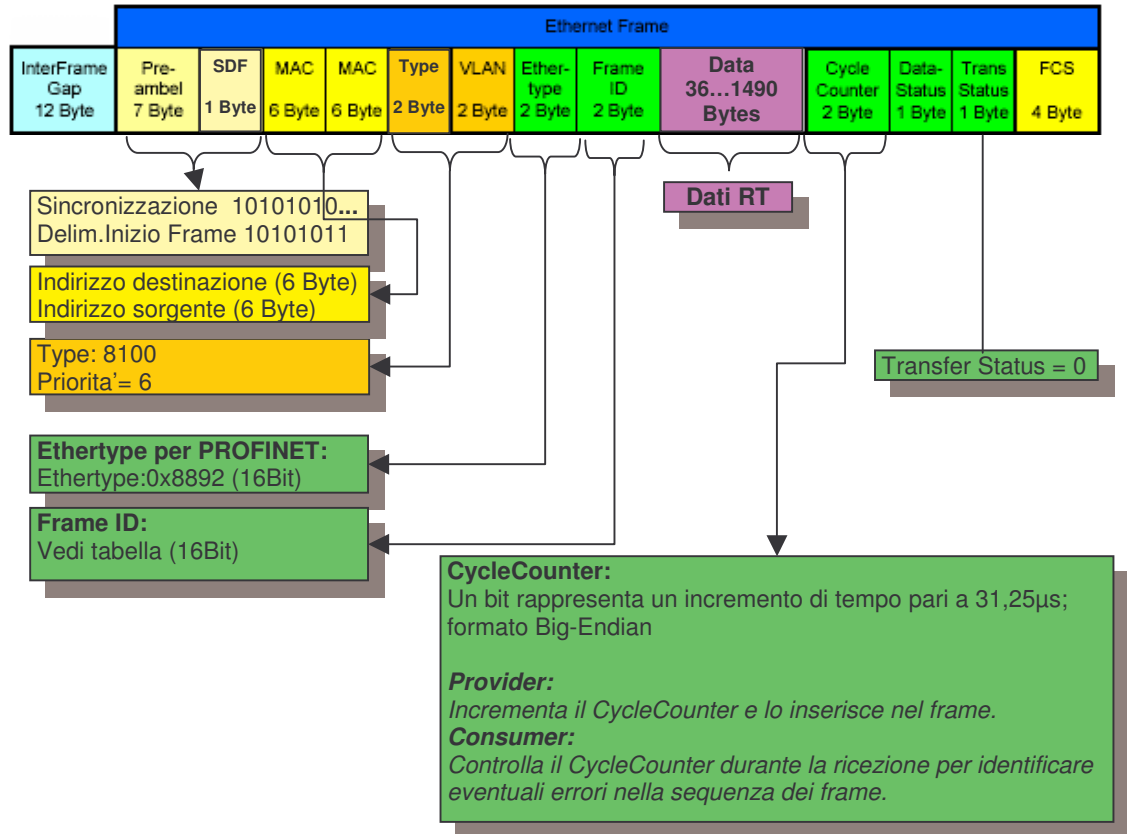



Figura 4.9 – Struttura del pacchetto Real-Time

FRAME ID 2 byte

Frame ID Da	 a	Significato
0000	00FF	Time Synchronization
0100	7FFF	RT classe 3 Frame ciclico (IRT)
8000	BFFF	RT classe 2 Frame ciclico (RT)
C000	FBFF	RT classe 1 Frame ciclico (RT)
FC00	FCFF	Trasmissione Aciclica "high" FC01: Allarme PN-IO, FC02: Evento PN
FD00	FDFF	riservato
FE00	FEFC	Trasmissione Aciclica „low" FE01: Allarme PN-IO, FE02: Evento PN
FEFD	FEFF	DCP
FF00	FFFF	riservato

L' EtherType 0x8892 identifica i pacchetti come pacchetti PROFINET real-time. Gli EtherType sono assegnati da IEEE e sono quindi univoci al fine di poter distinguere i vari protocolli.

Figura 4.10 – Analisi dei pacchetti tramite analizzatore di rete

Il *FrameID* è utilizzato per indirizzare un canale di comunicazione specifico tra due dispositivi. EtherType e FrameID fanno parte dell'header del protocollo RT e favoriscono una classificazione veloce dei pacchetti. Questo permette al dispositivo ricevente di classificare un pacchetto ricevuto velocemente per mezzo di un efficiente algoritmo. I 6-byte dell'indirizzo MAC selezionano il dispositivo. Il dispositivo real-time deve valutare solo l' EtherType e il FrameID per trovare il canale di comunicazione corretto.

Nel pacchetto viene anche inserito un contatore di ciclo "*Cycle Counter*" da parte del Provider. Il Provider incrementa il contatore di ciclo ad ogni ciclo e lo inserisce nel pacchetto. Ogni bit di incremento corrisponde ad un intervallo di tempo di 31,25 µs. Il Consumer, quando riceve il pacchetto, controlla il valore contenuto nel campo Cycle Counter. Attraverso questo valore può controllare se i dati di processo trasmessi sono quelli attuali.

I bit del campo *Data Status* hanno il seguente significato:

- Bit 0 (Stato) 1 = Primario; determina il canale dominante nel caso di canali multipli (ridondanza)
- Bit 2 (Dati Validi) 1 = Dati validi; il valore 0 è permesso solo durante la fase di avviamento
- Bit 4 (Stato del Processo) 1 = Il processo che ha generato i dati sta girando
- Bit 5 (Indicatore di Problema) 1 = Nessun problema; se questo bit è stato settato a 0, l'allarme diagnostico deve essere stato segnalato e le informazioni diagnostiche possono essere recuperate dai corrispondenti record.

Il byte *Transfer Status* deve essere uguale a 0.

4.2.5.2 OTTIMIZZAZIONE DEI DATI ATTRAVERSO L'USO DI PRIORITÀ (RT)

Per potere raggiungere un risultato ottimale, i pacchetti in PROFINET possono avvantaggiarsi delle priorità in conformità con IEEE 802.1q. se i pacchetti sono dotati di un VLAN Tag (Fig. 4.11) trasportano anche un valore di priorità codificato in un numero da 0 (senza priorità) a 7 (massima priorità).

I pacchetti che fluiscono fra i dispositivi sono gestiti dagli switch in base a questo concetto di priorità. Un pacchetto a più alta probabilità ha la precedenza durante la fase di accodamento all'interno dello switch sul pacchetto con priorità più bassa. L'uso della priorità basata su VLAN assicura che i pacchetti real-time siano preferiti, durante l'instradamento tramite switch, ai pacchetti standard del normale traffico IP. Ai pacchetti PROFINET real-time viene assegnato il valore 6.

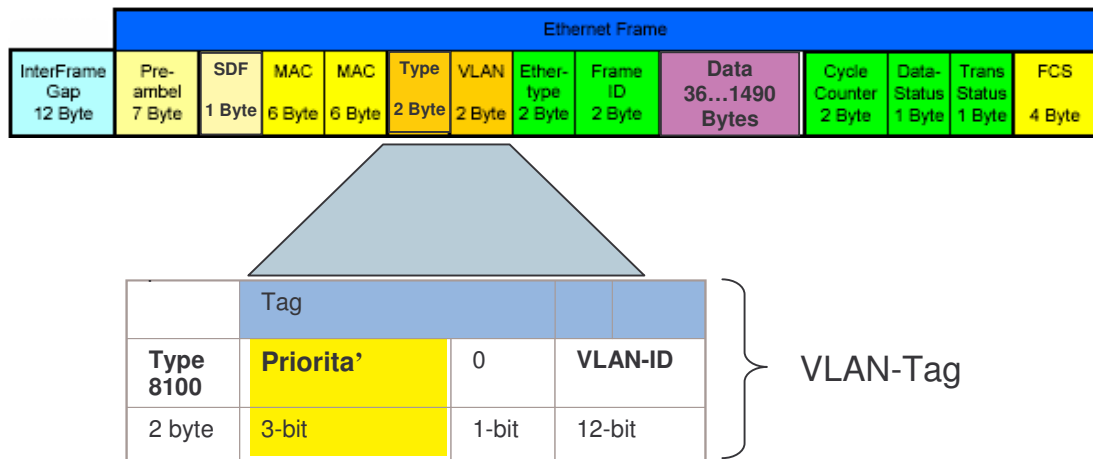



Figura 4.11 - Priorità

4.2.5.3 CICLO DI COMUNICAZIONE RT

Lo scopo delle suddivisone del ciclo di comunicazione RT è quello di dividere la banda all'interno del sistema. Per questo motivo non vengono definiti dei parametri di tolleranza per il jitter, parametri invece definiti nella comunicazione IRT.

I dispositivi PROFINET utilizzano una velocità di trasmissione di 100 Mbit/s in full-duplex e sono collegati tramite switch. In teoria, ciascun dispositivo potrebbe trasmettere i pacchetti usando tutta la banda massima, ma questa caratteristica porterebbe in breve tempo ad un cattivo funzionamento del sistema, infatti non sarebbe possibile che verso un singolo nodo siano diretti più flussi contemporanei tutti alla banda massima (overload) e neppure la trasmissione di pacchetti non RT (NRT). Gli switch interessati devono gestire le situazioni di sovraccarico attraverso delle code, oppure in ultima ipotesi gettare via i pacchetti.

Per questo motivo PROFINET prevede un meccanismo per gestire e limitare l'invio dei pacchetti. Per aumentare l'uso della banda e per prevenire sovraccarichi temporanei dovuti a connessioni multiple (ad esempio un PROFINET IO-Controller può essere collegato allo stesso tempo con diversi IO-Device) è possibile sincronizzare il ciclo di invio del provider con quello del consumer. Questa sincronizzazione è basata sul valore del contatore di ciclo "Cycle Counter" con una accuratezza stimata di un ciclo di clock.

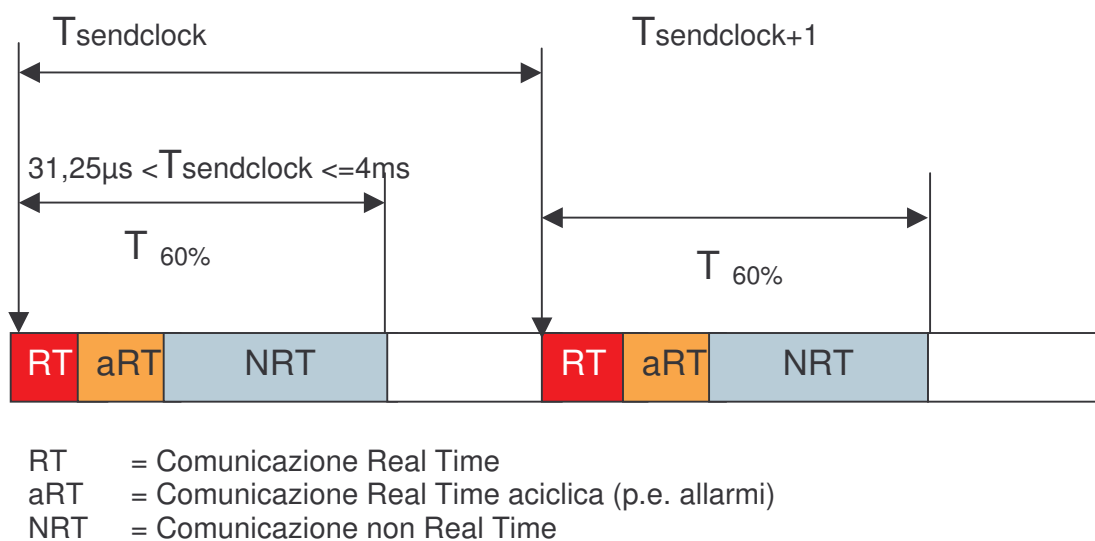



Figura 4.12 – Ciclo di comunicazione RT

Come mostrato in figura 4.12 il ciclo di comunicazione è suddiviso in parti differenti. Ogni ciclo è definito da $T_{sendclock}$ che è compreso tra 31,25 µs e 4 ms. All'inizio del ciclo vengono spediti i pacchetti RT relativi al dispositivo (RT). Il tempo destinato ai pacchetti RT non deve superare una certo ammontare (p.e. 50%) per ciascun ciclo. Successivamente possono essere spediti pacchetti real-time aciclici (aRT). Anche il tempo destinato per questi pacchetti non deve superare un certo ammontare (p.e. 10%). Infine, per il resto del tempo disponibile, verranno spediti i pacchetti non real-time (NRT) , p.e pacchetti UDP/TCP.

Attraverso il rapporto di riduzione "Reduction Ratio" e la fase possiamo distribuire i pacchetti RT nel tempo. Il traffico di rete viene così controllato e limitato attraverso questi meccanismi.

Questi meccanismi di gestione seguono le seguenti regole:

- il processo che si occupa dell'invio dei pacchetti non deve spedire alcun pacchetto RT se all'inizio del nuovo ciclo esiste una situazione di sovraccarico per l'interfaccia locale
- in media, non può essere usato più di un certo ammontare di banda per evitare il sovraccarico del sistema

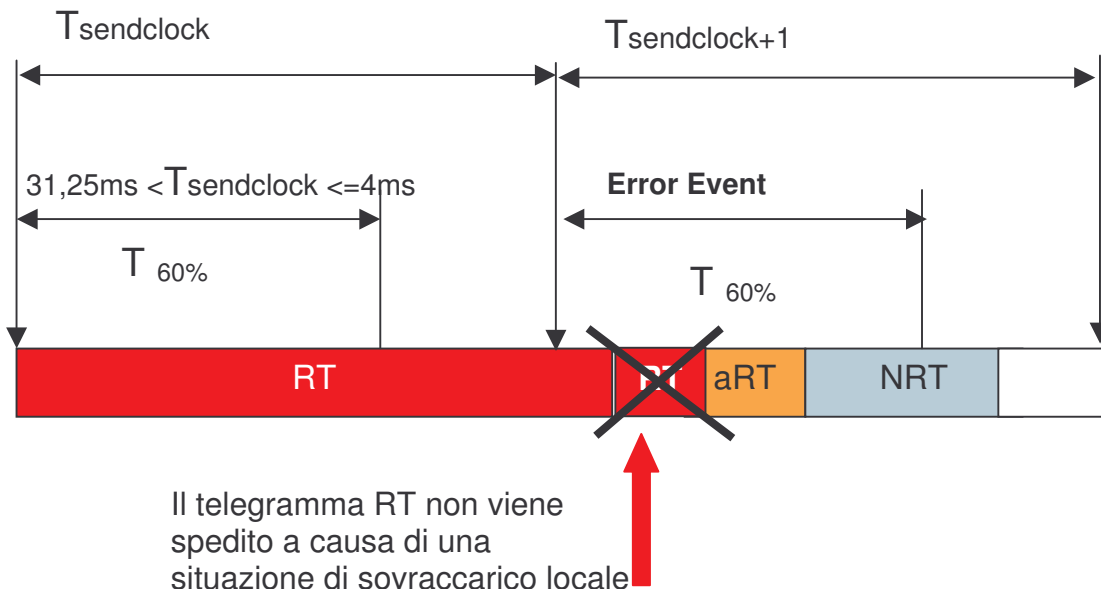



Figura 4.13 – Situazione di sovraccarico per l'interfaccia locale

La figura mostra l'esempio di una situazione di sovraccarico per l'interfaccia locale. Al tempo $T_{sendclock+1}$ il provider riceve un errore di trasmissione dal MAC dovuto ad una temporanea situazione di sovraccarico. In questo caso il provider salta completamente la spedizione dei pacchetti RT per l'intero ciclo e riprova con il prossimo ciclo. Tuttavia gli altri pacchetti (aRT e NRT) possono essere spediti all'interno del ciclo se questo è possibile.

4.2.6 COMUNICAZIONE ISOCRONA IN TEMPO REALE (IRT) IN PROFINET IO

Uso di applicazioni di motion control richiede la sincronizzazione dei cicli di applicazione. Tramite l'Isochronous Real-Time è possibile controllare cicli di bus di durata minore di 1 ms con jitter inferiore ad 1 μs.

Si tratta in questo caso di una soluzione hardware. In questo caso in PROFINET non c'è soltanto uno stack di comunicazione minimizzato, anche la trasmissione dei dati nella rete viene gestita attraverso particolari switch.

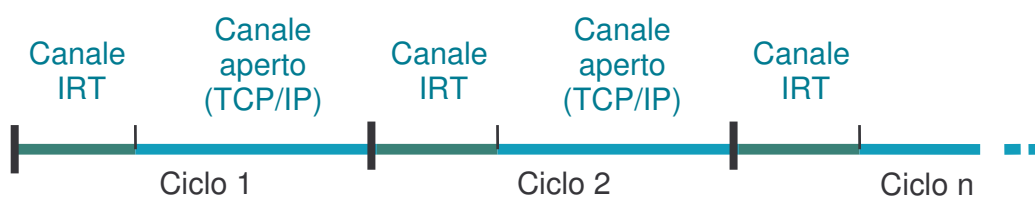



Figura 4.14 – Suddivisone temporale del ciclo di bus

In un sistema IRT il ciclo di bus deve essere diviso in una parte IRT e una parte non IRT durante la configurazione. PROFINET IRT usa gli stessi protocolli visti precedentemente; in aggiunta ha un protocollo di comunicazione per la sincronizzazione e un gestore della comunicazione temporale. *Lo scopo principale di IRT è la "sincronizzazione" della comunicazione attraverso un controllo preciso della comunicazione*.

Questo può essere implementato attraverso una programmazione della tempistica specifica per ogni singolo nodo inserito nella comunicazione. La programmazione della tempistica di comunicazione definisce esattamente quando un singolo pacchetto arriverà a destinazione e quando dovrà essere di nuovo instradato. Dovranno essere prese anche delle contromisure al fine di evitare che le stazioni senza comunicazioni IRT non interferiscano sul sistema.

4.2.6.1 SINCRONIZZAZIONE DI CLOCK.

La sincronizzazione del clock è disaccoppiata dai meccanismi IRT. La sincronizzazione usa le sequenze di protocollo secondo la normativa IEEE 1588 (PTP – Precision Time Protocol).

La sequenza della sincronizzazione del clock è descritta nel capitolo 2.2 “SINCRONIZZAZIONE DISTRIBUITA: IEEE1588”.

Il dispositivo che indica il tempo è chiamato PTP-Master e il dispositivo che riceve l’indicazione del tempo è chiamato PTP-Slave.

Tuttavia questo protocollo di sincronizzazione del tempo non è ottimale se nella rete sono presenti switch, perché una catena di switch può introdurre dei ritardi variabili influenzando le procedure di sincronizzazione della 1588, pensate per ritardi costanti. Per evitare questo problema, gli switch intermedi operano come PTP-Bridge. Questa è una procedura speciale sviluppata come una espansione della normativa IEEE 1588 che permette un alto grado di precisione in un sistema con switch real-time.

4.2.6.2 STRUTTURA DEL PACCHETTO PROFINET IRT

I pacchetti IRT sono soggetti ad una comunicazione guidata dal tempo. Una programmazione delle tempistiche nel dispositivo di campo IRT definisce il tempo esatto di trasferimento dei dati real-time. Gli altri pacchetti possono essere trasferiti solo dopo aver trasmesso i dati real-time.

Il pacchetto IRT è specificato dalla sua posizione temporale, dal Frame ID e dall’EtherType 0x8892.

Il campo VLAN per la priorità non è necessario e quindi non viene inserito nel pacchetto.

In figura è mostrata la struttura di un pacchetto IRT.

56 Bits	8 Bits	6 Byte	6 Byte	2 byte	2 byte	44..1496 bytes	4 byte
Preamb.	SYNCH	Indiriz. Dest.	Indiriz. Sorg.	Tipo 8892	Frame ID	IRT-Dati Utente	FCS

Figura 4.15 – Struttura del pacchetto PROFINET IRT

4.2.6.3 SUDDIVISIONE TEMPORALE DEL CANALE DI COMUNICAZIONE (IRT)

Utilizzando una rete Ethernet standard dotata di switch, i differenti pacchetti che sono stati spediti sulla stessa porta vengono processati secondo la strategia “il primo arrivato, è il primo servito”. Con il campo VLAN è possibile diminuire il ritardo di alcuni speciali pacchetti, ma l’accodamento può ancora sommare dei ritardi non prevedibili al tempo di trasferimento. Per ottenere il determinismo nel trasferimento di pacchetti Ethernet si deve riservare esplicitamente per i pacchetti IRT una percentuale della banda disponibile, non solo nei dispositivi, ma anche negli switch.

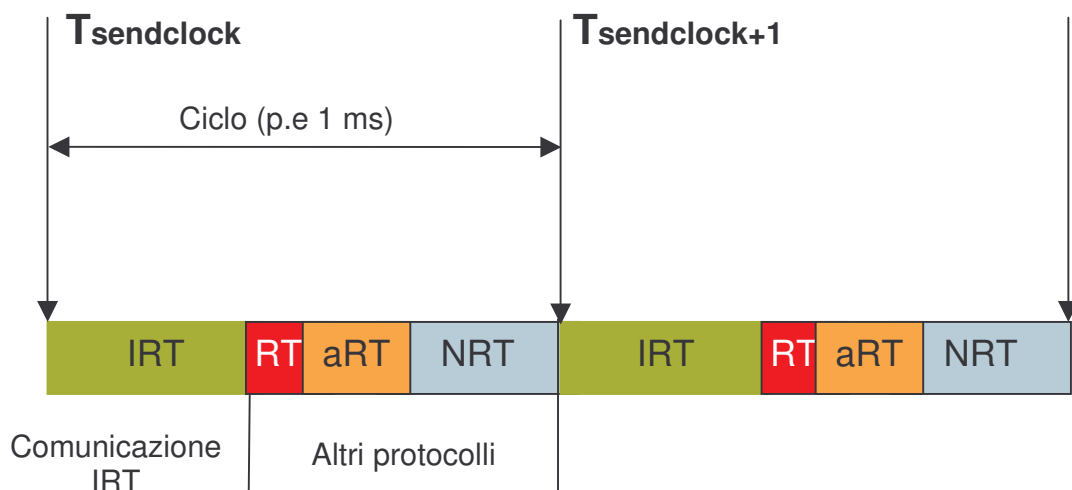



Figura 4.16 – Suddivisone temporale del canale di comunicazione

La banda disponibile è divisa in due sezioni, banda per la comunicazione IRT e banda per le altre comunicazioni.
All'inizio del ciclo è inserita la comunicazione IRT. All'interno del tempo destinato all'IRT possono essere trasmessi solo pacchetti IRT. I pacchetti IRT sono identificati dal loro Frame ID. Nel frattempo gli altri pacchetti sono inseriti in un buffer fino all'inizio della parte non IRT. Nella parte non IRT i pacchetti sono spediti in base alle regole definite nella IEEE 802.1 D. Come descritto in precedenza, per l'invio dei pacchetti vengono valutate le rispettive priorità.
Il jitter di sincronizzazione di un ciclo IRT deve essere inferiore ad 1 microsecondo. Per ottenere un trasferimento di dati real-time deterministico all'interno di un ciclo IRT il tempo di invio di ogni pacchetto è calcolato esplicitamente ed è fisso per ogni pacchetto. Parimenti vengono definiti i limiti del ciclo IRT. I limiti degli altri pacchetti (RT, aRT, NRT), per la parte del ciclo non IRT, dipendono dal numero di pacchetti da spedire e possono variare nel tempo. I limiti del ciclo IRT (inizio e fine) possono essere utilizzati come indicatori per l'inizio di una applicazione (p.e. motion control)

4.2.6.4 PIANIFICAZIONE DELLA TEMPISTICA IRT BASATA SULLA TOPOLOGIA D'IMPIANTO

Per il trasferimento dei dati attraverso i protocolli TCP/IP o UDP/IP è sufficiente configurare i dispositivi che devono comunicare (producer, consumer). Tutti gli altri componenti come gli switch che si trovano fra i dispositivi hanno delle tabelle di indirizzi che vengono aggiornate durante la comunicazione.
Diversamente IRT ha bisogno non solo di conoscere i dispositivi che devono comunicare, ma anche tutti i nodi che si trovano sul tragitto. La conoscenza dell'esatta topologia porta i seguenti vantaggi:

- Garanzia del determinismo: Per ogni pacchetto il tempo di invio ad un dispositivo è definito esattamente e il tempo di invio di un pacchetto è diverso da quello degli altri pacchetti dello stesso nodo. Inoltre pacchetti spediti su porte diverse possono essere spediti allo stesso istante di tempo. Sono definiti anche i tempi di spedizione di tutti i nodi con switch che si trovano tra il

producer e il consumer. Non avviene l'accodamento dei pacchetti IRT e così il trasferimento dei dati all'interno dello switch può essere ridotto al minimo.

- Ottimizzazione dell'uso della banda: Pacchetti destinati a parti differenti del sistema possono essere trasferiti allo stesso istante di tempo. L'algoritmo che pianifica le temporizzazioni può suggerire topologie alternative per l'ottimizzazione.

4.2.6.5 CONFIGURAZIONE DI UNA RETE IRT

La configurazione delle temporizzazioni avviene tramite un algoritmo.
Questo algoritmo necessita dei seguenti input:

- Topologia della rete e dati delle prestazioni per ogni nodo
- Indirizzo del nodo sorgente e del nodo destinatario
- Quantità dei dati trasmessi per ogni producer o consumer
- Caratteristiche della connessione quali tipo (fibra, rame, WLAN), lunghezza, ritardo ecc.

La topologia è definita come la connessione fisica delle porte degli switch contenute nella rete. Tuttavia, la topologia include anche il tempo di transito di un pacchetto attraverso lo switch e la velocità di trasferimento dei dati sul mezzo fisico.
L'algoritmo restituisce i seguenti dati di output relativi ad ogni trasmissione ed a ogni switch:

- Porta ricevente in cui il pacchetto arriva e una o più porte trasmittenti per il suo reinoltro
- Tempo esatto di trasferimento
- Fase di un ciclo di comunicazione in cui inserire il pacchetto quando viene utilizzato il "Reduced Ratio"

Per la pianificazione delle temporizzazioni è utilizzato un servizio centralizzato, l'algoritmo di pianificazione viene fatto girare sul sistema di ingegnerizzazione. In questo modo l'ottimizzazione delle temporizzazioni di un sistema composto, per esempio, da 100 stazioni può essere realizzata tramite un personal computer di prestazioni medie in pochi secondi.

Comparato con un sistema con comunicazione tramite indirizzi, il risultato di questa pianificazione comporta l'aggiunta di parametri addizionali che devono essere caricati nelle varie stazioni.

4.2.6.6 INTEGRAZIONE DI UNO SWITCH NEI DISPOSITIVI PROFINET IO

Per sfruttare al massimo le prestazioni di IRT sono stati progettati degli ASIC con controllore Ethernet con funzionalità di switch e funzionalità IRT integrate. Questi nuovi ASIC permettono diverse opzioni di cablaggio per interfacciare dispositivi di campo con Ethernet tra cui l'economica struttura a linea tipica degli impianti PROFIBUS (fig. 4.17). Questo permette una riduzione dei costi considerevole, perché in molti casi non è necessario l'utilizzo di switch esterni, e il cablaggio di un sistema di automazione in una struttura a linea permette una notevole riduzione dei costi dei cavi e del loro posizionamento.

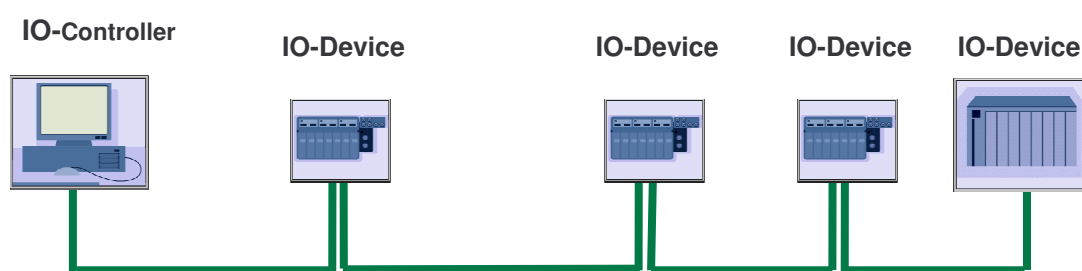



Figura 4.17 – Struttura a linea

Attualmente Siemens offre due ASIC che possono essere sia ASIC Ethernet quanto ASIC PROFINET: ERTEC 400 e ERTEC 200.

L'ERTEC 400 (*Enhanced Real-Time Ethernet Controller*) è un controllore Ethernet ad alte prestazioni con integrato uno switch a 4 porte e un microprocessore per applicazioni industriali a 32-bit, l'ARM 946.

L'ERTEC 200 è funzionalmente identico ma prevede una struttura con 2 porte.

4.2.7 CALCOLO DELLE PRESTAZIONI DI UN SISTEMA PROFINET IRT

Per valutare le prestazioni di un sistema PROFINET IRT è stata analizzata un'applicazione in un impianto di produzione realizzata tramite una rete di comunicazione con topologia lineare dotata di un IO-Controller e 31 IO-Device.
Il numero di device è stato scelto per poter comparare i risultati calcolati nel capitolo 3 (§ 3.2). Valuteremo quindi il tempo di aggiornamento del sistema (tempo necessario alla trasmissione delle uscite e all'acquisizione degli ingressi del sistema).
Il tempo calcolato fa riferimento solo allo scambio di dati real-time isocroni, per calcolare il tempo totale del ciclo isocrono dovranno essere sommati i tempi di comunicazione riservati allo scambio di dati non isocrono (comunicazione RT, comunicazione asincrona, ecc.), che sono impostabili in fase di configurazione del sistema.

4.2.7.1 TEMPO DI TRASMISSIONE DATI DI OUTPUT A TUTTI I DISPOSITIVI E TEMPO DI AGGIORNAMENTO DI UN SISTEMA PROFINET

Il tempo di trasmissione dipende essenzialmente dai seguenti fattori:

- Velocità di trasmissione dei dati
- Tipo di trasmissione (Full Duplex)
- Tempo di propagazione
- Topologia della rete
- La lunghezza dei dati di Output o di Input (minimo 44 byte)

Per lo studio del sistema analizziamo in che modo vengono scambiati i dati tra i vari dispositivi IRT. La figura 4.18 mostra il principio di funzionamento dello scambio dati tra dispositivi di campo IRT con switch integrato.
Tutti i pacchetti possono essere messi strettamente in sequenza calcolando esattamente il loro tempo di invio. Se l'intera comunicazione va o viene da un nodo (IO-Controller), allora il tratto di linea con il carico più pesante si trova sul tratto che connette questo nodo e il suo adiacente.
Siccome PROFINET opera in full-duplex, i dati di output e di input possono essere trasferiti simultaneamente.
Analizzando l'invio di dati da parte del IO-controller agli IO-device vediamo che al tempo t1 viene posto sul bus il pacchetto indirizzato al dispositivo #4, seguito dagli altri pacchetti. Al tempo t2 il pacchetto #4 è transitato all'interno dello switch del dispositivo #1 ed è stato instradato sul tratto di linea che connette il dispositivo successivo. Al tempo t3 il pacchetto indirizzato al dispositivo #3 è transitato all'interno dello switch del dispositivo #1 ed è stato instradato sul tratto di linea che connette il dispositivo successivo. Il pacchetto #4 è transitato all'interno dello switch del dispositivo #2 ed è stato instradato sul tratto di linea che connette il dispositivo successivo. Tutto questo continua fino al raggiungimento dei vari dispositivi da parte dei pacchetti corrispondenti. Questo è un esempio di come sono instradati i pacchetti, il tempo di arrivo dipende da vari fattori e un pacchetto può giungere a destinazione prima di un altro.

Invio dati di Output:

tempo
t4
t3
t2
t1
AI #1
AI #2
AI #3
AI #4
AI #2
AI #3
AI #4
AI #3
AI #4
AI #4

IO-Controller
IO-Device #1
IO-Device #2
IO-Device #3
IO-Device #4

FULL DUPLEX

Invio dati di Input:

tempo
t1
t2
t3
t4
DaI #1
DaI #2
DaI #3
DaI #4
DaI #2
DaI #3
DaI #4
DaI #3
DaI #4
DaI #4

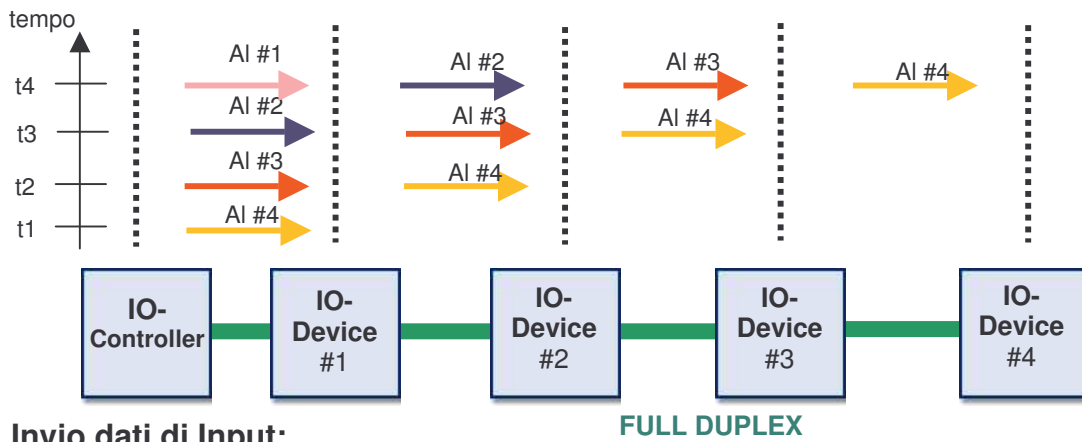
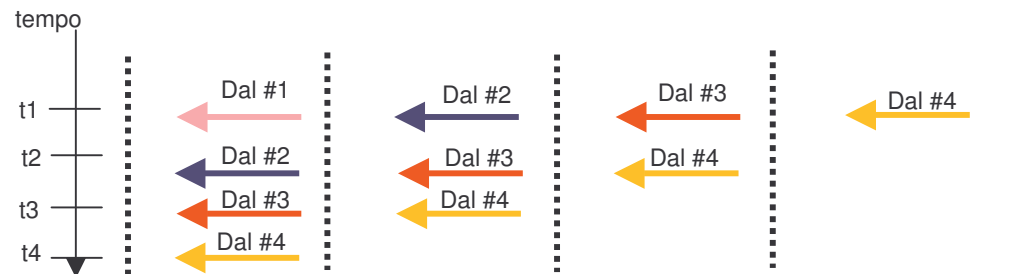



Figura 4.18 – Trasferimento dei pacchetti attraverso gli switch in PROFINET IRT

Per calcolare la quantità di dati trasferiti bisogna considerare che la lunghezza minima di un pacchetto è di 64 byte. Sommati a questi ci sono poi 12 byte di idle e 8 byte di preambolo (includendo il byte di inizio frame). In totale questo porta la lunghezza minima del pacchetto IRT uguale a 84.

Lunghezza del campo dati minima è quindi di 44 byte. Questa quantità di dati è normalmente sufficiente per la maggior parte delle applicazioni con scambio di dati ciclico.

Per ottimizzare il tempo di aggiornamento il primo pacchetto immesso sul bus è quello diretto all'ultimo IO_Device.

Per calcolo dei tempi di trasmissione per raggiungere n-esimo IO-Controller ho considerato i seguenti parametri:

$$N_{bit-pacchetto_i} = N_{bit-IDLE} + N_{bit-PREAMB} + N_{bit-HEADER} + N_{byte-pacchetto_i} \cdot 8 + N_{bit-TRAILER}$$

N_s = numero totale degli slave

v_{Tra} = velocità di trasmissione

$T_{t\text{-}switch}$ = tempo di transito attraverso lo switch del dispositivo

L = lunghezza del segmento

v_{Prop} = velocità di propagazione minima pari a 175000000 m/s

Per cui il tempo di trasmissione risulta:

$$T_{Tra_n} = \sum_{i=n}^{N_S} \left(N_{bit-pacchetto_i} \frac{1}{v_{Tra}} \right) + \sum_{i=1}^{n} \left(T_{t-switch_i} + \frac{L_i}{v_{Prop}} \right)$$

Il tempo di risposta del sistema, essendo la trasmissione full-duplex è dato dal valore massimo tra il tempo di trasmissione degli Output e degli Input:

$$T_{AGG_PROFINET_IO} = \max\left(T_{Tra_n}\right)$$

Il calcolo dei tempi di trasmissione per raggiungere n-esimo IO-Controller sono stati calcolati scrivendo un programma in linguaggio Matlab e utilizzando la seguente formula matriciale:

$$T_{Tra} = \left(A \cdot N_{bit-pacchetto}\right)\frac{1}{v_{Tra}} + B \cdot \left(T_{t-switch} + L \cdot \frac{1}{v_{Prop}} \right)$$

Dove

$$A = \begin{bmatrix} 1 & 1 & 1 & . \\ 0 & 1 & 1 & . \\ 0 & 0 & 1 & . \\ . & . & . & . \end{bmatrix}$$ matrice quadrata triangolare superiore di dimensioni $N_S x N_S$

$$A = \begin{bmatrix} 1 & 0 & 0 & . \\ 1 & 1 & 0 & . \\ 1 & 1 & 1 & . \\ . & . & . & . \end{bmatrix}$$ matrice quadrata triangolare inferiore di dimensioni $N_S x N_S$

$N_{bit\text{-}pacchetto}$ = vettore colonna di lunghezza N_S contente i bit del pacchetto da inviare ad ogni nodo

$T_{t\text{-}switch}$ = vettore colonna di lunghezza N_S contente i tempo di transito attraverso lo switch del dispositivo

L = vettore colonna di lunghezza N_S contente le lunghezze da dispositivio a dispositivo

T_{Tra} = vettore colonna di lunghezza N_S contenente il tempo necessario per raggiungere ogni nodo

Appendice A

IL MODELLO ISO/OSI

A.1 DESCRIZIONE DEL MODELLO

Ogni qualvolta si parla di comunicazione in termini tecnici, si fa riferimento al "modello ISO/OSI". Di seguito viene fornita una descrizione del modello che ci servirà per studiare il funzionamento dei vari protocolli isocroni.

Questo modello si basa su di una "proposta" sviluppata dall'organizzazione internazionale degli standard (*International Standard Organization*: **ISO**) come primo passo verso la standardizzazione internazionale dei vari protocolli. Esso è denominato **modello di riferimento OSI dell'ISO**; la sigla OSI è l'acronimo di *Open System Interconnection*.

Il modello OSI è caratterizzato da sette strati che furono stabiliti in base ai seguenti principi:

- Deve essere creato uno strato laddove è richiesto un differente livello di astrazione.
- Ciascun strato deve svolgere una funzione ben definita.
- La funzione di ciascun strato deve essere scelta tenendo conto della definizione dei protocolli standard internazionali.
- I confini tra gli strati devono essere scelti in modo da minimizzare il flusso di informazioni attraverso le interfacce.
- Il numero di strati deve essere abbastanza grande, da non permettere il raggruppamento di funzioni distinte nel medesimo strato se non in caso di necessità, e abbastanza piccolo da non appesantire l'architettura.

I bus di campo implementano tipicamente un ulteriore livello denominato **Strato Utente** per le strategie di controllo distribuite, che li diversificano ulteriormente delle tradizionali reti di comunicazione.

Il modello OSI di per se non è un'architettura di rete perché non specifica esattamente i servizi ed i protocolli da utilizzare in ciascun strato, ma si limita a definire ciò che ciascuno strato dovrebbe fare. La funzione di modello, serve quindi come semplice punto di riferimento e confronto, ma soprattutto non rappresenta una norma alla quale ogni sistema deve assolutamente aderire.

In generale non è indispensabile disporre di tutti e sette i livelli; a seconda delle funzionalità necessarie qualche livello intermedio è spesso omesso. Infatti nei bus di campo vengono implementati tipicamente il primo, il secondo ed il settimo livello per ragioni di efficienza.

Applicazione
Presentazione
Sessione
Trasporto
Rete
Collegamento dati
Fisico

Figura A.1 - Lo stack ISO/OSI

A.1.1 LIVELLO 1: LO STRATO FISICO

Lo **strato fisico** riguarda la trasmissione di bit lungo un canale di comunicazione. Gli aspetti del progetto riguardano il meccanismo per garantire che un bit "1" trasmesso da un'estremità venga ricevuto come tale (e non come "0") all'altra estremità.

Le problematiche tipiche da affrontare in questo contesto riguardano i livelli di tensione o di corrente usati per rappresentare il bit 1 o il bit 0, la durata in microsecondi di un bit, la possibilità di trasmettere simultaneamente in entrambi le direzioni, il modo in cui viene stabilita la connessione iniziale ed il modo in cui viene disattivata quando entrambe le estremità hanno terminato, il numero di "piedini" che deve avere il connettore di rete e la funzione di ciascun piedino.
Qui gli aspetti di progetto hanno a che fare in gran parte con le interfacce meccaniche, elettriche e procedurali, e col mezzo fisico di trasmissione, situato al di sotto dello strato fisico.

A.1.2 LIVELLO 2: COLLEGAMENTO DATI

Il compito principale dello **strato di collegamento dati** (*Data Link Layer*) è quello di prendere un mezzo fisico "rudimentale" e di trasformarlo in una linea esente, per quanto possibile, da errori di trasmissione per lo strato di rete. Esso svolge questo compito facendo si che il trasmettitore organizzi i dati di input in *frame*, li trasmetta sequenzialmente ed elabori i *frame di riconoscimento* inviati in risposta dal ricevitore.
Poiché lo strato fisico si limita ad accettare e trasmettere una serie di bit senza interessarsi del loro significato o della loro struttura, sarà lo strato di collegamento dati a dover creare e riconoscere i confini dei frame, il che può essere fatto aggiungendo speciali configurazioni di bit all'inizio ed alla fine di ogni frame.
Lo strato di collegamento dati può essere ulteriormente suddiviso in due sottostrati; quello inferiore che si interfaccia allo strato fisico e chiamato **MAC** (*Media Access Control*), quello superiore che si interfaccia allo strato di rete e chiamato **LLC** (*Logical Link Control*)
Un impulso di rumore sulla linea può distruggere completamente un frame. In questo caso il software dello strato di collegamento dati sulla macchina di provenienza dovrà ritrasmettere il frame. E' questo strato che ha il compito di risolvere problemi causati da messaggi danneggiati, persi o duplicati.

Un'altra questione che si presenta allo strato di collegamento dati (come pure nella maggior parte degli altri strati) è il modo in cui impedire ad un trasmettitore veloce di sommergere di dati un ricevitore lento.

A.1.3 LIVELLO 3: RETE

Lo **strato di rete** (*Network Layer*) riguarda il controllo del funzionamento della sottorete. Una questione fondamentale di progetto è la determinazione del modo in cui i pacchetti sono instradati dall'estremità di provenienza a quella di destinazione.
Se troppi pacchetti sono presenti contemporaneamente nella sottorete, essi si intralceranno a vicenda, formando un "collo di bottiglia". Anche il controllo di una congestione è sotto la responsabilità dello strato di rete.
Quando un pacchetto deve viaggiare da una rete all'altra per raggiungere la sua destinazione, possono sorgere molti problemi: l'indirizzamento usato dalla seconda rete può differire da quello della prima, la seconda rete potrebbe addirittura non accettare il pacchetto perché è troppo grande, i protocolli potrebbero essere diversi, e così via. E' lo strato di rete che ha il compito di risolvere questi problemi per consentire l'interconnessione di reti omogenee.

A.1.4 LIVELLO 4: TRASPORTO

La funzione fondamentale dello **strato di trasporto** (*Tansport Layer*) consiste nell'accettare i dati dallo strato di sessione, nel suddividerli in unità più piccole (se necessario), nel passare queste allo strato di rete e nel garantire che i pezzi arrivino tutti correttamente all'altra estremità. Inoltre il tutto deve essere svolto efficientemente ed in modo da isolare lo strato di sessione dalle inevitabili modifiche nella tecnologia dell'hardware.
Lo strato di trasporto determina anche il tipo di servizio da fornire allo strato di sessione e, in definitiva, agli utenti della rete. Il tipo più diffuso di connessione di trasporto è un canale da punto a punto, esente da errori, che fornisce i messaggi nell'ordine in cui sono stati inviati. Comunque altri tipi di servizio sono rappresentati

dal trasporto di messaggi isolati senza garanzia sull'ordine di consegna e dal broadcasting di messaggi a più destinatari.
Lo strato di trasporto ha anche il compito di stabilire e cancellare le connessioni attraverso la rete. Ciò richiede un meccanismo di denominazione, affinché il processo presente su una macchina abbia il modo per descrivere il destinatario con cui deve conversare.

A.1.5 LIVELLO 5: SESSIONE

Lo **strato di sessione** (*Session Layer*) permette agli utenti su macchine differenti di stabilire tra loro delle sessioni. Una sessione consente l'ordinario trasporto dei dati, come fa lo strato di trasporto, ma fornisce anche dei servizi potenziati utili in certe applicazioni. Una sessione potrebbe servire ad un utente per connettersi ad un sistema remoto in time-sharing, o per trasferire un file tra più macchine.
Uno dei servizi dello strato di sessione è la gestione del controllo del dialogo. Le sessioni possono permettere che il traffico si svolga contemporaneamente nelle due direzioni, oppure in una sola direzione alla volta. Se il traffico è unidirezionale lo strato di sessione può servire a tenere nota dei turni.
Servizi di sessione attinenti sono la *gestione del token,* per evitare che diversi nodi eseguano operazioni nello stesso istante, e la *sincronizzazione* tra i processi, ad esempio inserendo "punti di controllo" durante il trasferimento di dati, per evitare, in caso di errori, di ritrasmettere tutto il messaggio.

A.1.6 LIVELLO 6: PRESENTAZIONE

Lo **strato di presentazione** (*Presentation Layer*) esegue certe funzioni che sono richieste così spesso da rendere opportuna per esse una soluzione generale. In particolare, diversamente da tutti gli strati inferiori che sono interessati soltanto al trasferimento affidabile di bit da un punto all'altro, lo strato di presentazione ha a che fare con la sintesi e la semantica delle informazioni trasmesse. Al fine di rendere

possibile la comunicazione tra computer con rappresentazioni simboliche differenti, le strutture dati da scambiare possono essere definite in modo astratto, insieme con una codifica standard da utilizzare sulla linea di comunicazione. Il compito di gestire queste strutture dati astratte e di effettuare la conversione dalla rappresentazione usata all'interno del computer alla rappresentazione standard della rete è gestito dallo strato di presentazione.
Questo strato ha a che fare con altri aspetti della rappresentazione delle informazioni come ad esempio la compressione dei dati e la crittografia.

A.1.7 LIVELLO 7: APPLICAZIONE

Lo **strato applicazione** (*Application Layer*) contiene vari protocolli che sono necessari per offrire vari servizi agli utenti, quali ad esempio:
- terminale virtuale;
- trasferimento file;
- posta elettronica;

In generale il settimo livello concerne la descrizione dell'interfaccia verso l'applicazione; tuttavia questo strato non comprende l'applicazione stessa, in quanto essa non fa parte del sistema di comunicazione.

Appendice B

PROFIBUS DP
(DEVICE PERIPHERAL)

B.1 IL PROTOCOLLO PROFIBUS

Profibus (PROcess FIeldBUS) è un bus di campo fortemente voluto da Siemens, il cui sviluppo ed amministrazione è stato ufficialmente affidato ad un'organizzazione denominata Profibus Trade Organization (PTO), composta da membri provenienti dal settore costruttivo, da istituti di ricerca e dall'utenza finale. Attualmente vi sono anche altre organizzazioni per lo sviluppo di Profibus, quali Profibus User Organization (PNO) e Profibus International (PI); per l'Italia esiste il Profibus Network Italia (PNI) con sede a Brescia.

La tecnologia PROFIBUS è nata come standard tedesco secondo la normativa DIN 19245, successivamente è stata riconosciuta conforme allo standard europeo (EN 50170), e insieme ad altri bus di campo è stata poi standardizzata dalle norme internazionali IEC 61158. E' utilizzabile per una vasta gamma di applicazioni industriali, e consente a dispositivi di diversi produttori di comunicare tra loro senza particolari aggiustamenti di interfaccia.

Il successo di PROFIBUS, che può vantare una crescita vertiginosa in pochi anni, è dovuto al suo ampio spettro di applicabilità; dalla semplice connessione di attuatori e sensori fino al livello di cella, dall'industria manifatturiera al settore macchine utensili, dal settore del building al processo fino all'industria chimica e petrolchimica.

E' un bus di campo in grado di utilizzare, con lo stesso mezzo fisico, servizi per connessioni semplici orientate al byte e servizi per connessioni complesse (frame di configurazione, di parametrizzazione, di diagnostica ecc.) in modalità ciclica e

aciclica, è in grado di consentire la coesistenza di diversi dispositivi che implementano quanto richiesto dalle varie applicazioni sul mercato.

La famiglia PROFIBUS consiste di tre protocolli compatibili tra loro:

- **PROFIBUS FMS** (Fieldbus Message Specification): rappresenta la soluzione "general-purpose" per la comunicazione anche a livello di cella. Il protocollo FMS mette a disposizione potenti servizi per una vasta gamma di applicazioni e consente grande flessibilità. E' stata la prima implementazione ed oggi è stata praticamente abbandonata.

- **PROFIBUS DP** (Device Peripheral): ottimizzato per collegamenti economici ad alte prestazioni. Questa versione PROFIBUS è rivolta soprattutto alla comunicazione tra sistemi di controllo dell'automazione e dispositivi distribuiti di I/O.

- **PROFIBUS PA** (Process Automation): progettato principalmente per l'automazione di processo, consente di collegare sensori ed attuatori su una linea di comunicazione comune in aree a sicurezza intrinseca (EExi). Con il protocollo PA è possibile trasmettere dati e alimentazione su un bus a due conduttori in accordo con lo standard internazionale IEC 1158-2.

B.2 ARCHITETTURA DEL PROTOCOLLO

I protocolli PROFIBUS si basano sul modello di riferimento OSI (Open System Interconnection) di ISO (International Standard Organization) secondo il quale ogni strato del protocollo gestisce compiti definiti. In particolare il Livello 1, Physical Layer (PHY) definisce le caratteristiche fisiche della trasmissione, il Livello 2, Data Link Layer (Profibus utilizza il nome Fielbus Data Link, FDL) definisce il protocollo di accesso al bus e il Livello 7, Application Layer descrive l'interfaccia verso l'applicazione. (Fig. B.1)

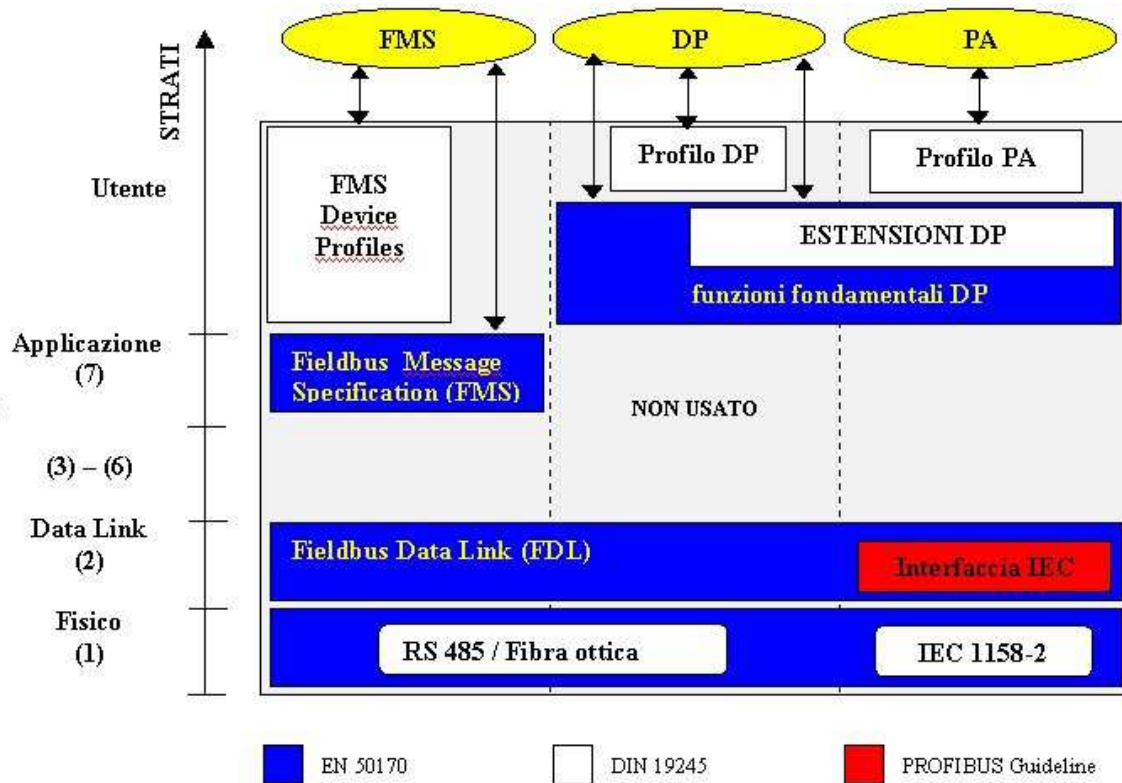



Figura B.1 – Architettura dei protocolli PROFIBUS

B.3 MODELLO DI COMUNICAZIONE

Le caratteristiche tecniche e funzionali di PROFIBUS si fondano sull'implementazione di un bus seriale con cui dispositivi digitali decentralizzati possono essere collegati assieme a livello di sensori/attuatori, di campo e di cella.
Tutte le versioni di PROFIBUS (DP, FMS e PA) utilizzano un protocollo uniforme di accesso al bus implementato dal livello 2 del modello ISO/OSI (Fig. B.1).
Per quanto riguarda i dispositivi, PROFIBUS distingue tra:

- Dispositivi MASTER, che controllano la comunicazione sul bus. Un master può spedire messaggi senza richiesta esterna quando detiene il controllo della linea di comunicazione (possesso del token). Vengono anche chiamati stazioni attive.

- Dispositivi SLAVE: sono unità periferiche che non possono accedere al bus direttamente, se non per divulgare informazioni diagnostiche che lo riguardano. Possono solamente riconoscere messaggi ricevuti o spedire messaggi al master se richiesti esplicitamente. Dal momento che richiedono piccole porzioni di protocollo, la loro implementazione è particolarmente economica. Vengono anche chiamate stazioni passive.

La gestione dell'accesso al bus viene implementata nel protocollo PROFIBUS secondo la filosofia master/slave, mentre per quanto riguarda l'accesso al bus tra master, si utilizza una procedura basata su token passing.
La procedura token passing garantisce che il permesso di accesso al bus, concesso al master che possiede il token, sia definito per un intervallo di tempo preciso e costante; quindi il tempo che ogni master dovrà attendere per accedere al bus dipenderà dal numero di dispositivi attivi presenti nella rete e dal massimo tempo di utilizzo del bus permesso (token hold time). Il token viene passato da un master all'altro secondo un ordine prefissato (anello logico).
La procedura master/slave consente alla stazione attiva che in un preciso momento possiede il token di accedere alle stazioni passive a lui assegnate. Il master può spedire messaggi agli slave o richiedere messaggi dagli slave. In questo modo si possono implementare configurazioni:

- master/slave puri (o mono-master)
- master/master puri
- combinazioni delle due configurazioni precedenti

Durante la fase di inizializzazione del sistema basato sul bus di campo, il compito del sottostrato MAC (Medium Access Control, sottolivello del livello 2) di ogni dispositivo master è di definire il proprio indirizzo all'interno dell'anello logico costituito da tutti i dispositivi attivi. Durante la fase operativa, i master malfunzionanti o spenti devono essere tolti dall'anello logico, mentre quelli attivati successivamente devono essere inseriti nell'anello logico.
Un altro compito importante del livello 2 (FDL) riguarda la sicurezza dei dati. Il formato delle trame utilizzato nel protocollo PROFIBUS assicura elevata integrità

dei dati grazie a telegrammi caratterizzati da distanza di Hamming HD=4 ottenuti secondo le norme dello standard IEC 870 5-1.

Lo strato di collegamento dati permette il trasferimento delle informazioni anche in modalità broadcasting, utilizzata per l'invio di trame a tutte le stazioni attestate sul bus, e multicasting per l'invio di dati ad un gruppo di stazioni.

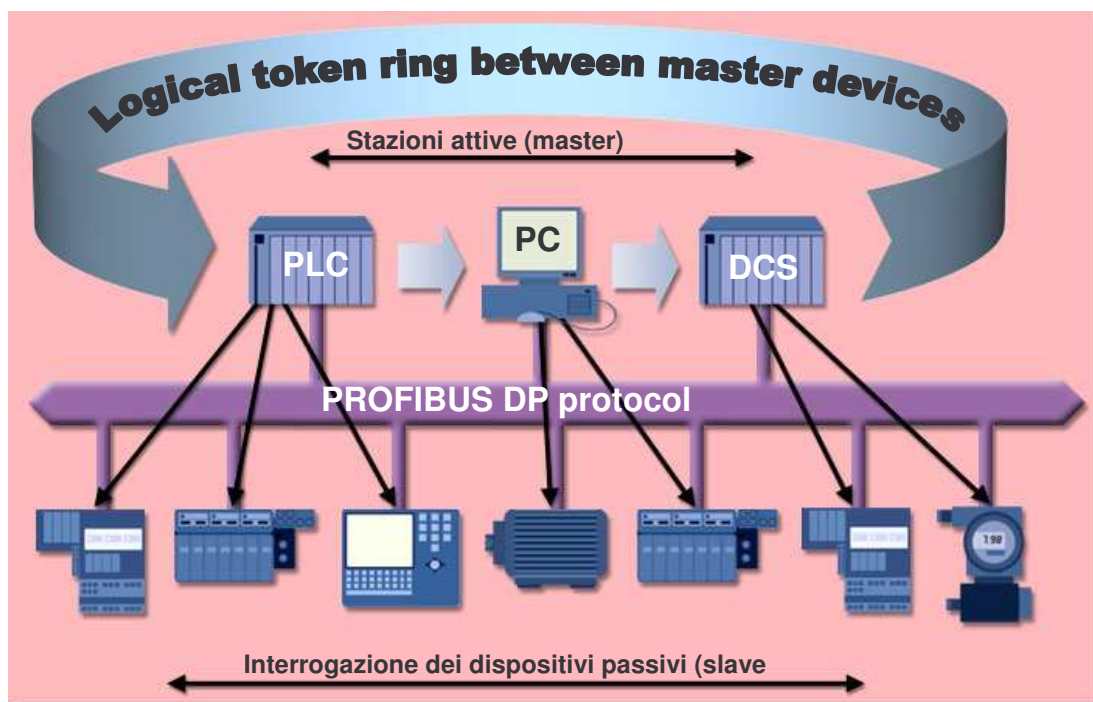



Figura B.2 – Protocollo di accesso al bus, comune alle tre versioni PROFIBUS

B.4 PROFIBUS DP - CARATTERISTICHE DI BASE

La versione Device Peripheral (DP) del protocollo PROFIBUS è stata pensata per la comunicazione tra sistemi di controllo dell'automazione (i.e. PC e PLC) e dispositivi di ingresso/uscita distribuiti (i.e. sensori/attuatori) ed è ottimizzata per comunicazioni ad alte velocità e per connessioni poco costose. La maggior parte dello scambio di dati in questo contesto avviene in modo ciclico, tuttavia per permettere lo svolgimento di procedure di configurazione, di diagnostica o di gestione degli allarmi, il protocollo supporta anche funzioni di comunicazione aciclica.

Sono state rilasciate tre versioni del protocollo Versione 0, Versione 1, Versione 2.

Le caratteristiche si base nate come protocollo PROFIBUS DP sono state inserite nel protocollo rinominato PROFIBUS DP-V0 (PROFIBUS DP Versione 0).

Tutte queste caratteristiche sono disponibile anche nelle successive versioni del protocollo. Per questo motivo si inizia la trattazione dalla versione DP-V0 infatti molte delle sue funzionalità saranno utili per comprendere il funzionamento della versione isocrona di PROFIBUS (PROFIBUS DP-V2).

La tabelle che segue illustra gli aspetti più salienti di PROFIBUS DP-V0.

Tecnologia di trasmissione: • RS 485 con doppio conduttore attorcigliato (eventualmente schermato) o fibra ottica • Velocità di trasmissione da 9.6 Kbit/s a 12 Mbit/s
Accesso al bus: • Token passing tra master, master/slave tra dispositivi attivi e passivi • Possibili configurazioni mono-master e multi-master • Dispositivi master e slave, massimo 126 stazioni per ogni segmento compatibilmente con la tecnologia di trasmissione.
Comunicazione: • Peer to Peer per dati utente, Multicast per comandi di controllo • Ciclica (trasmissione dati master/slave e master/master)

Modalità operative (DPM1): • *Operate*: Trasferimento ciclico di dati in ingresso ed in uscita • *Clear*: gli ingressi dei DP slaves vengono letti, ad ogni uscita viene assegnato valore “0” • *Stop*: è permessa la sola trasmissione di dati tra master e master • *Off-line*: nessuna forma di trasmissione permessa
Sincronizzazione: • Comandi di controllo consentono la sincronizzazione degli ingressi e delle uscite • Modalità *Sync*: le uscite sono sincronizzate • Modalità *Freeze*: gli ingressi sono sincronizzati
Funzionalità: • Trasmissione ciclica di dati utente tra DP master e DP slave • Attivazione e disattivazione dinamica di singoli DP slave • Verifica della configurazione dei DP slave • Potenti funzioni diagnostiche (3 livelli gerarchici di messaggi diagnostici) • Sincronizzazione degli ingressi e/o delle uscite • Assegnamento degli indirizzi ai DP slaves sul bus • Configurazione dei DP master (DPM1) sul bus per mezzo dei DP master (DPM2) • Massimo 246 byte di dati in ingresso e in uscita per DP slave
Funzioni di protezione e sicurezza: • Tutti i messaggi sono trasmessi con distanza di Hamming HD = 4 • Watchdog timer per i dispositivi DP passivi • Protezione d’accesso in ingresso e in uscita per Master DPM1 e per Slave. • Funzioni di controllo dei dati utente trasmessi (Data_Control_Timer)
Tipi di dispositivi: • DP master di classe 2 (DPM2): dispositivi di programmazione, configurazione, diagnostici • DP master di classe 1 (DPM1): controllori programmabili come PLCs, PCs, ecc. • DP slave: dispositivi con ingressi e uscite analogiche o digitali, trasmettitori ecc.

Tab. B.1 – Caratteristiche generali del protocollo DP-V0

Comunque, la velocità elevata di trasmissione non rappresenta l'unico elemento che ha determinato il successo della versione DP del protocollo PROFIBUS; procedure di installazione e servizi semplici, potenzialità diagnostiche e trasmissioni senza errori sono caratteristiche altrettanto importanti per gli utilizzatori.

B.4.1 CONFIGURAZIONE DEL SISTEMA E TIPOLOGIA DEI DISPOSITIVI

La descrizione della configurazione del sistema è data dall'indicazione del numero di stazioni impiegate, dall'assegnamento degli indirizzi, dal formato dei messaggi di input/output, dal formato dei messaggi diagnostici e dai parametri del bus.
Il protocollo PROFIBUS DP, in particolare, permette sistemi mono-master e multi.master, e questo aspetto garantisce un alto grado di libertà durante la configurazione del sistema. Si possono utilizzare fino a 126 dispositivi (master o slave) connessi ad un bus ed è permesso uno scambio massimo di 246 byte di dati sia in ingresso che in uscita.
Come illustrato in tabella B.1 ogni sistema PROFIBUS DP può contenere dispositivi di una delle tre seguenti classi:

- DP Master class 1 **(DPM1)**

E' un controllore centrale che scambia informazioni con le stazioni decentrate ad esso assegnate (i.e. DP slave), in un ben definito ciclo di messaggi. Gestisce la trasmissione dei dati utente e può comunicare con i dispositivi master DPM2.
Tipicamente appartengono a questa classe PLC, PC o sistemi VME.

- DP Master class 2 **(DPM2)**

E' un dispositivo di programmazione utilizzato durante la fase di allestimento per configurare e monitorare il sistema.
Appartengono a questa classe i dispositivi di configurazione, come i personal computer, ed i pannelli di controllo .

- **DP Slave**

E' un dispositivo periferico (dispositivo di input/output, pilota, valvola, Human Machine Interface) che riceve dati in ingresso (richieste) ed invia informazioni in uscita al controllore centrale (risposte).
Uno slave DP è indirizzabile da entrambi i tipi di master.

Profibus DP gestisce sistemi mono-master/slave o multi-master/slave e implementa tipiche forme di comunicazione uno a uno o uno a molti. Può essere utilizzata una modalità di comunicazione multicast (uno a molti), solo per diffondere comandi di controllo globali da parte di entrambi i tipi di master verso un gruppo di slave DP, o da parte di un dispositivo DPM2 verso un gruppo di DPM1.
In una comunicazione di tipo master/slave è il master (DPM1 o DPM2) a possedere il controllo, mentre in una comunicazione master/master è il dispositivo DPM2 e gestire l'interazione. Non è definita nessuna forma di comunicazione tra dispositivi della stessa categoria.
La quantità di informazione scambiata tra stazioni dipende dai tipi di dispositivi in comunicazione; al massimo si possono trasferire 246 byte di dati sia in input che in output.
Nei sistemi mono-master (Fig. B.3) un solo master è attivo durante la fase operativa. Solo in questo tipo di configurazione è possibile ottenere i più corti cicli di bus permessi dal protocollo PROFIBUS DP, quindi le prestazioni migliori in termini di velocità di trasferimento dati.
Possono essere collegati fino a 125 slave DP in una rete.

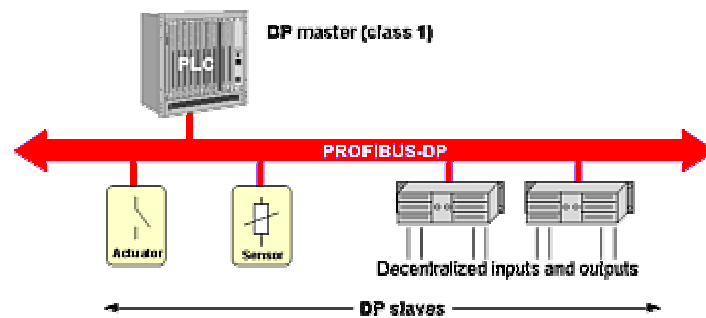



Fig. B.3 - Struttura di un sistema mono-master

Nei sistemi multi-master (Fig. B.4) più stazioni attive sono collegate al bus in fase operativa. L'accesso al mezzo è consentito alla sola stazione attiva che possiede il token, il quale viene trasmesso tra i master presenti in rete lungo un anello logico, definito in fase di inizializzazione del sistema. Ciascun master, insieme agli slave che può indirizzare, rappresenta un sottosistema indipendente. C'è da precisare che gli ingressi e le uscite degli slave possono essere lette da qualsiasi master. Le restrizioni esistono solo in fase di scrittura, operazione permessa al solo master (DPM1) a cui uno specifico slave è stato assegnato durante la configurazione di sistema. Le prestazioni non eguagliano quelle ottenibili con una configurazione mono-master, soprattutto se il numero di master supera le tre unità. E' possibile collegare fino a 126 dispositivi tra master e slave in una rete.

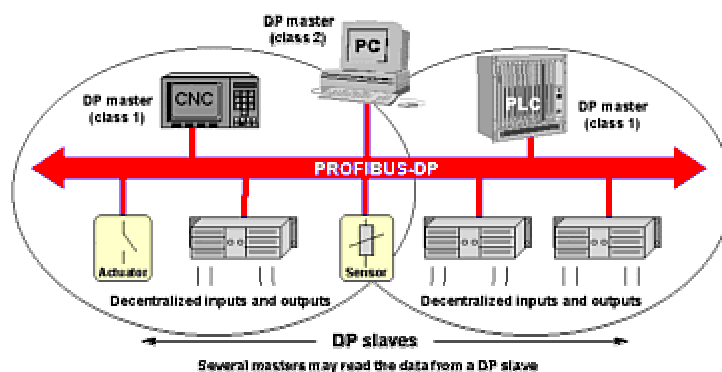



Fig. B.4 - Struttura di un sistema multi-master

Tutte le stazioni di un sistema PROFIBUS DP devono aver assegnato un indirizzo sul bus. Per i dispositivi DP slave tale indirizzo potrà essere definito anche via bus dai soli dispositivi DPM2.

Quando un dispositivo passivo non possiede ancora un proprio indirizzo gli viene assegnato provvisoriamente il 126 (di default). Attraverso tale indirizzo una stazione DPM2 può successivamente accedere al DP slave e modificare il valore identificativo con uno alternativo al 126. I dispositivi attivi non possono avere un indirizzo di default. Per motivi di sicurezza un dispositivo DPM1 non può scambiare dati con un dispositivo DP slave indirizzato col 126.

B.4.2 PROCEDURA TRASMISSIVA

Lo scambio di messaggi avviene in cicli. Ogni ciclo di messaggio consiste in un frame di richiesta da parte del master e dal corrispondente frame di conferma (che non contiene dati utente, es. scrittura) o di risposta (che contiene dati utente, es. lettura) da parte della stazione indirizzata (master o slave). Un ciclo di messaggio può essere interrotto dalla trasmissione del token o di un comando di controllo globale (broadcast).
Tutte le stazioni, fatta eccezione per quella che detiene il token, verificano che i messaggi presenti sul bus siano a loro indirizzati. La conferma o la risposta del destinatario deve ritornare al mittente entro un intervallo di tempo prestabilito TSL (Slot Time). Se ciò non avviene, il master richiedente potrà ritrasmettere il frame, allo scadere di un intervallo di tempo TID (Idle Time), per un numero predefinito di volte. Se, dopo questi ulteriori tentativi di comunicazione, il dispositivo indirizzato non risponde, viene automaticamente segnato come non operativo. Quando un master deve comunicare con un dispositivo non operativo, trasmette il messaggio un'unica volta senza tentativi successivi. In caso di conferma o risposta, la stazione indirizzata ritorna ad essere considerata operativa.

PROFIBUS DP prevede in generale diversi tipi di trasmissioni che riguardano:

- La gestione del token.
- La registrazione delle stazioni.
- La trasmissione ciclica (PROFIBUS DP-V0).
- La gestione della diagnostica e degli allarmi.
- La trasmissione aciclica (PROFIBUS DP-V1).
- La trasmissione isocrona (PROFIBUS DP-V2).
- La trasmissione da slave a slave (PROFIBUS DP-V2).

B.4.3 TOKEN PASSING

Il token è un frame speciale che viene trasmesso tra dispositivi master seguendo un ordine crescente di indirizzo. Ovviamente la stazione caratterizzata dal più alto identificatore, trasmetterà il token a quella con l'indirizzo più basso.
Ricezione del token: ogni master possiede una lista di tutte le stazioni attive LAS (List of Active Station) nella quale sono segnati, fra gli altri, il proprio indirizzo TS (This Station) e quelli dei dispositivi immediatamente precedente PS (Previous Station) e successivo NS (Next Station). Se un master riceve per la prima volta il token da una stazione diversa da PS, viene prodotto un messaggio di errore; ma se la cosa si ripete nel ciclo di messaggio immediatamente successivo, allora il destinatario assume che l'anello logico sia stato modificato e memorizza la stazione mittente come nuove PS.
Trasmissione del token: dopo aver concluso il proprio ciclo di messaggio, il master che detiene il token lo passa alla stazione attiva successiva (NS). A trasmissione avvenuta e trascorso un certo intervallo temporale TSYN (Syn Time, all'interno dello Slot Time), il mittente può ricevere o un messaggio di convalida (valid frame) a garanzia del buon esito dell'operazione, oppure può riceverne uno alternativo (invalid frame), dal quale desume che un altro master (DPM2) ha preso nel frattempo il controllo del bus.
L'azione successiva da parte del master che inizialmente aveva trasmesso il token, è di verificare l'effettivo utilizzo della linea di comunicazione, il tutto all'interno dello stesso Slot Time iniziale. In caso di 'bus idle' il mittente replica l'intera procedura al massimo per altre due volte e se i risultati non cambiano spedisce il token alla stazione successiva a quella individuata da NT nella LAS. Se per tutte le stazione presenti nella LAS non succede nulla, allora vuol dire che il master considerato è l'unico nell'anello logico.

B.4.4 TEMPO DI POSSESSO DEL TOKEN

L’intervallo temporale in base al quale ogni stazione attiva è autorizzata a mantenere il possesso del token TTH (Token Holding Time) è individuato dalla differenza di altri due parametri: TTR (Target Rotation Time) e TRR (Real Rotation Time)

TTH = TTR - TRR

TRR è la differenza temporale tra l’ultima e la penultima volta in cui il DPM1 considerato ha ricevuto il token.
TTR è un valore predefinito in fase di inizializzazione e per quanto riguarda i dispositivi DPM1 obbedisce alla seguente espressione:
TTR (DPM1) = TSLP (TS) + TTR (DPM2) + TADD
dove:
TSLP(TS) : tempo assegnato al ciclo di polling relativo al dispositivo considerato TS.
TTR(DPM2) : Target Rotation Time riferito ai dispositivi DPM2 (> (TSLP).
TADD : tempo considerato come margine di sicurezza per eventuali ritrasmissioni dei messaggi.
Determinato il tempo di possesso del token TTH, esso viene decrementato ogni Tbit (tempo necessario per trasmettere un bit: 1/bit rate). Terminato un ciclo di messaggio, il master che detiene il token potrà proseguire nell’utilizzo del bus solamente se TTH>0.

B.4.5 INSERIMENTO E RIMOZIONE DI STAZIONI DALL'ANELLO LOGICO

Stazioni master e/o slave possono essere connesse/disconnesse al/dal mezzo trasmissivo in ogni momento. Ciascun dispositivo attivo è responsabile degli indirizzi compresi tra TS (This Station) e NS (Next Station) denominati GAP e rappresentati nella GAP List (GAPL). Ogni GAP viene controllato periodicamente all'interno di un intervallo temporale TGUD (GAP Update Time) per mezzo del frame 'Request FDL Status' (operazione con cui ogni stazione attiva verifica i dispositivi presenti in rete). La fase di manutenzione del GAP inizia al ricevimento del token dopo la trasmissione di tutti i messaggi in coda, se il TTH disponibile lo consente. Altrimenti si rimanda al ricevimento del token successivo dopo la trasmissione dei messaggi ad alta priorità.
Gli indirizzi appartenenti al GAP vengono analizzati in ordine ascendente. Se una stazione risponde come 'not ready' o 'slave station', viene segnata come tale nella GAPL e si procede al controllo dell'indirizzo successivo. Se una stazione attiva risponde come pronta ad entrare nell'anello logico, il dispositivo attivo che detiene il token deve segnarla nella GAPL come NS, quindi deve passarle il controllo del bus. Nel caso in cui una stazione passiva appartenente alla GAPL non risponde a ripetuti 'Request FDL Status', viene rimossa dalla lista.

B.4.6 INIZIALIZZAZIONE DELL'ANELLO LOGICO

La fase di inizializzazione dell'anello logico parte dalla stazione attiva con l'indirizzo più basso. Tale master DP trasmette due token frame a se stesso (NS=TS) con i quali informa tutti gli altri dispositivi attivi che è l'unica stazione nell'anello logico. Successivamente, seguendo gli indirizzi in ordine crescente, spedisce ad ognuno di questi una 'Request FDL Status'. Se una stazione risponde come 'master non pronto' o come 'slave', viene inserita nella GAPL, se risponde come 'master pronto ad entrare nell'anello logico', viene registrata come NS nella LAS (definendo così anche il GAP) e gli viene trasmesso il token frame.

B.4.7 PRIORITA' DEI MESSAGGI

Il protocollo PROFIBUS DP prevede l'utilizzo di messaggi a bassa priorità LP (Low Priority) e ad alta priorità HP (High Priority). Tutti i messaggi di richiesta nella comunicazione di tipo master/slave sono spediti con priorità alta, così come i messaggi di risposta a nuove informazioni diagnostiche da parte di dispositivi passivi. In tutti gli altri casi i messaggi sono trasmessi a bassa priorità.
Quando una stazione master riceve il token, spedisce prima tutti i messaggi HP poi quelli LP. Se inizialmente capita che TRR sia maggiore di TTR, quindi TTH<0, può essere spedito comunque un messaggio HP inclusi ulteriori tentativi in caso di errori, dopo di che il token deve essere passato alla stazione master successiva NS.

B.4.8 COMPORTAMENTO DI SISTEMA DEI MASTER DPM1

Le specifiche PROFIBUS DP includono una descrizione dettagliata del comportamento del sistema, allo scopo di garantire un corretto scambio di informazioni tra i dispositivi. Tale comportamento è legato allo stato operativo dell'interfaccia utente dei dispositivi DPM1, ciascuno dei quali può essere controllato localmente o via bus da un modulo master DPM2. Esistono quattro stati possibili per i DPM1:

- Stop: i parametri del bus sono caricati nel Fieldbus Data Link (FDL), ma nessuna forma di comunicazione tra DPM1 e slave DP è permessa (solo tra DPM1 e DPM2).
- Clear: Stato in cui il master DPM1 prova a parametrizzare e configurare le stazioni passive, può leggere gli ingressi dei DP slave, ma deve mantenere le uscite in uno stato di 'fail-safe' (tutte pari a '0').
- Operate: DPM1 è in fase di trasferimento dati. In un ciclo di comunicazione ogni DPM1 può leggere i dati in ingresso dei DP slave e scrivere le informazioni di uscita degli stessi. Se l'interfaccia utente lascia lo stato Operate, le uscite di tutti gli slave saranno annullate con un comando di controllo globale.
- Off-line: non è permessa qualsiasi forma di comunicazione tra dispositivi. Il controller FDL locale viene estromesso dall'anello logico dei master.

Ciclicamente, ogni DPM1 spedisce informazioni circa il suo stato agli slave DP assegnati con un comando multicast.
La reazione del sistema a un errore, durante la fase trasmissiva di un DPM1, dipende dal valore assunto da un parametro denominato Error_Action_Flag (parametro del bus presente a livello di User Interface nei DPM1). Se a questo parametro è stato assegnato il valore '1', il DPM1 porta le uscite di tutti i DP slave assegnati nello stato di fail-safe e modifica il proprio stato in Clear. Se il parametro Error_Action_Flag ha valore '0', DPM1 rimane nello stato Operate e sarà l'utente a definire il comportamento del sistema.

B.4.9 COMUNICAZIONE CICLICA TRA MASTER DPM1 E SLAVE (MS0)

Lo scambio di dati ciclico tra i master di classe1 DPM1 e uno slave è anche chiamato connessione MS0 (Master/Slave PROFIBUS DP-V0)
La trasmissione dati tra DPM1 e DP slave è gestita dal dispositivo master attivo (fig. 3.5), secondo un ordine definito dall'utente durante la fase di progettazione del sistema. Sempre in questa fase, l'utente specifica quali DP slave devono essere assegnati ai DPM1 e quali devono essere inclusi o esclusi dalla trasmissione ciclica col master.
Ogni dispositivo master DPM1 deve possedere una base di dati contenente parametri e valori di configurazione (es. tipo di dispositivo, formato e lunghezza delle informazioni, numero di ingressi e di uscite, ecc.) relativi ai dispositivi passivi a lui assegnati ed al bus.
Durante la fase di parametrizzazione e configurazione, ciascuno slave DP confronta le informazioni ricevute dal dispositivo attivo con quelle proprie memorizzando i risultati sulla diagnostica interna. Il master DPM1 verifica la compatibilità tra le due versioni richiedendo le informazioni diagnostiche allo slave e solamente dopo una positiva conclusione di quest'ultimo controllo è possibile passare alla successiva fase di trasferimento dati.
Quindi, ricapitolando, se una stazione master DP vuole comunicare con un dispositivo slave DP, deve innanzitutto verificarne lo stato attuale (es. se il dispositivo è collegato in rete, o se è occupato da un DPM2) attraverso la richiesta

delle informazioni diagnostiche (funzione Slave_Diag). Se non emerge alcun problema, si passa alle successive fasi di parametrizzazione (funzione Set_Prm) e di verifica dei dati di configurazione (funzione Chk_Cfg), al termine delle quali dovranno essere richieste nuovamente le informazioni diagnostiche per verificare se sono presenti errori di parametrizzazione e/o configurazione o se lo slave DP è già occupato con un altro master DP oppure se lo slave DP non è ancora pronto per comunicare. Nei primi due casi, il master DP ripete la procedura dall'inizio, nel terzo richiede le ultime informazioni diagnostiche finché il messaggio non scompare. Se non si verificano errori, può iniziare lo scambio dei dati tra master e slave (funzione Data_Exchange).
Oltre al trasferimento di dati utente, ogni dispositivo DPM1 può spedire, in qualsiasi momento e a qualunque stazione passiva a lui assegnata, nuovi dati di parametrizzazione senza uscire dalla modalità 'Operate'. Inoltre qualunque master DPM2 può leggere in ogni momento gli ingressi , le uscite e le informazioni diagnostiche di ciascuno slave DP.

B.4.10 DIAGRAMMA DI STATO DELL'INTERFACCIA UTENTE DI UNO SLAVE DP-V0

La figura B.5 riporta il diagramma degli stati che descrivono il funzionamento di un dispositivo slave DP-V0.
All'accensione lo slave si trova nello stato "POWER_ON" dal quale passa allo stato "WAIT_PARAM" dopo che è stato fissato il suo indirizzo di rete. In questo stato si attende l'arrivo del messaggio di parametrizzazione da parte del master con il quale vengono fissati i parametri caratteristici del funzionamento dello slave.
Nello stato "WAIT_CONFIG" lo slave attende il messaggio di configurazione, il quale fornisce il numero ed il tipo di ingressi/uscite che verranno scambiati nella fase successiva. Se la configurazione è accettata lo slave entra nello stato "DATA_EXCH" che rappresenta la fase di normale operatività durante la quale i dati di processo vengono ciclicamente scambiati con il master.

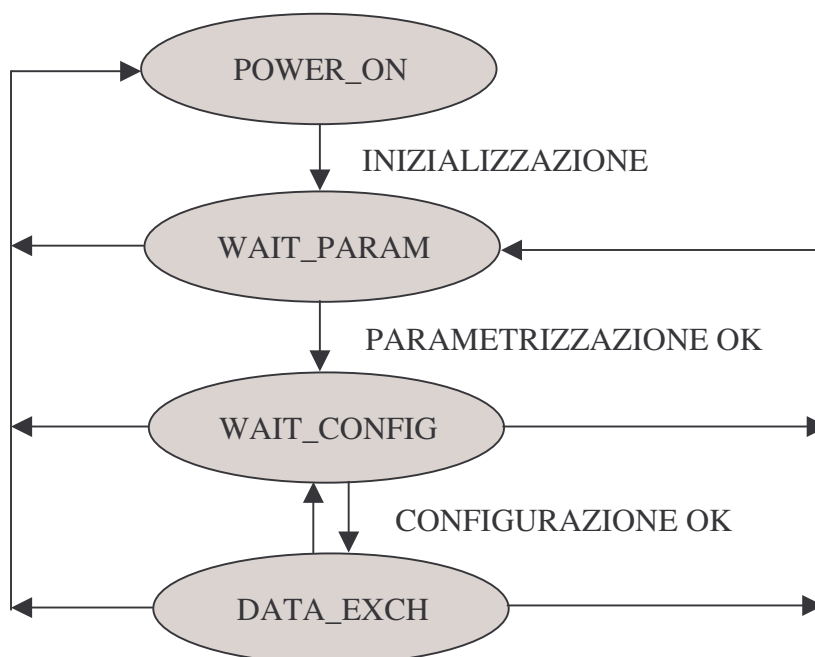



Figura B.5 - Diagramma di stato di uno slave DP-V0

B.4.11 SINCRONIZZAZIONE DEGLI SLAVE DP-V0 (COMANDI SYNC E FREEZE)

Oltre a trasferire dati utente alle stazioni collegate, ogni master può spedire comandi di controllo globali a singoli DP slave, a gruppi di DP slave (multicast) o a tutti i DP slave contemporaneamente (broadcast). Questi comandi di controllo, tra le altre finalità, consentono di attivare la modalità di comunicazione 'sync' e 'freeze' con i dispositivi DP slave per sincronizzare il loro funzionamento.

Gli slave attivano la modalità di funzionamento sync quando ricevono il comando omonimo dal loro master. In questa fase le uscite di tutti gli slave indirizzate vengono bloccate nel loro stato corrente e rimangono invariate finché un nuovo comando sync non verrà inviato. La modalità sync termina al ricevimento del comando unsync.

Allo stesso modo gli slave assumono modalità di funzionamento freeze quando ricevono il comando omonimo dal loro master. In questa fase operativa, lo stato degli ingressi viene bloccato al valore corrente e l'aggiornamento non è consentito finché

un nuovo comando freeze non verrà inviato. La modalità freeze termina al ricevimento del comando unfreeze.

B.4.12 FUNZIONI DIAGNOSTICHE

Il protocollo Profibus DP prevede l'utilizzo, da parte delle stazioni attive e passive, di particolari servizi che permettono di rilevare velocemente errori e malfunzionamenti. In una tipica configurazione master/slave, i messaggi diagnostici sono trasmessi sul bus dal dispositivo passivo e raccolti da quello attivo.
La comunicazione delle informazioni diagnostiche rappresenta l'unico caso di autonomia trasmissiva concessa agli slave dal protocollo DP. Più precisamente, uno slave DP può trasmettere fino a 6 byte di diagnostica standard, che contengono informazioni di stato comuni per tutti gli slave. Sono inoltre utilizzabili 26 byte (estensibili a 238) di diagnostica estesa, i quali possono riportare dati specifici, determinati dall'utente in base alle proprie esigenze. L'area di memoria rivolta alla diagnostica estesa è organizzata in tre livelli:

- Diagnostiche riferite alle stazioni: questi messaggi riguardano in generale lo stato operativo di tutte le stazioni (eccessi di calore, basso voltaggio, ecc.)
- Diagnostiche riferite ai moduli: nel caso in cui uno slave DP sia a struttura modulare, questi messaggi indicano se un funzionamento scorretto si sia verificato in un modulo o in uno specifico sotto insieme di ingressi o uscite di un dispositivo passivo.
- Diagnostiche riferite al canale: questi messaggi individuano un errore in uno specifico bit di I/O.

B.4.13 MECCANISMI DI PROTEZIONE

Al fine di garantire una comunicazione affidabile e sicura il protocollo PROFIBUS DP offre funzioni di protezione da errori di parametrizzazione e 'failure' della comunicazione. Esse si basano su meccanismi di watchdog abilitati sia sul versante master che su quello slave.
Le stazioni attive utilizzano un timer (Data_Control_Timer) per ogni stazione slave ad esso assegnata, che scatta quando la comunicazione con lo slave corrispondente non ha luogo entro un certo intervallo di tempo. Quando un timer si attiva l'utente viene informato e se l'opzione di reazione automatica del sistema agli errori è stata attivata, il master DPM1 che si trova nello stato Operate, porta le uscite di tutti gli slave assegnati a '0' e modifica il proprio stato in Clear.
Le stazioni passive parimenti controllano che la comunicazione con la stazione attiva abbia luogo entro un intervallo di tempo stabilito e se questo non accade portano le proprie uscite a '0'.
In aggiunta a questi meccanismi che controllano l'attività delle differenti stazioni, lo standard garantisce la protezione degli accessi agli ingressi e alle uscite degli slave nella configurazione multi-master; questo fa si che solamente le stazioni master abilitate possano eseguire un accesso diretto alle uscite e agli ingressi delle stazioni slave ad essi assegnate, tutti gli altri moduli master possono solamente consultare una immagine degli ingressi e delle uscite che non possono alterare.

B.4.14 COMUNICAZIONE ACICLICA (PROFIBUS DP-V1)

Diversamente dalla comunicazione ciclica lo scambio dati ciclico avviene solo quando è necessario.
La comunicazione aciclica tra i master di classe1 DPM1 e uno slave è anche chiamata connessione MS1 (Master/Slave PROFIBUS DP-V1).
Lo scambio di dati ciclico (DP-V0) permette solo di indirizzare il dispositivo di campo nel suo complesso. La versione del protocollo DP-V1 permette di indirizzare i singoli moduli di uno slave. Questo permette, ad esempio, di cambiare il tipo di

misura o i valori di fondo scala durante il processo produttivo attraverso la parametrizzazione dei singoli moduli.

La comunicazione aciclica e' normalmente una comunicazione non critica dal punto di vista del tempo. Quindi lo scambio dati aciclico avviene dopo lo scambio di dati ciclico. Nel trasferimento dati ciclico, DP-V0, i dati sono trasferiti con brevi telegrammi all'interno del ciclo di bus. In DP-V0 la lunghezza del telegramma è sempre collegata alla lunghezza dei dati utente che devono essere trasferiti. Quindi, procedendo all'ottimizzazione di questa lunghezza a seconda del processo si può ottenere un sostanziale aumento delle prestazioni.

I servizi aciclici per DP-V1 hanno portato ad una estensione della struttura sei telegrammi DP-V0. Sono stati perciò definiti nuovi Service Access Points (SAPs) che implementano il nuovo meccanismo di indirizzamento per lo scambio dei dati utente (Function_Num ,Slot_Number, index, lenght).

La figura mostra la struttura di un ciclo di bus in DP-V1.

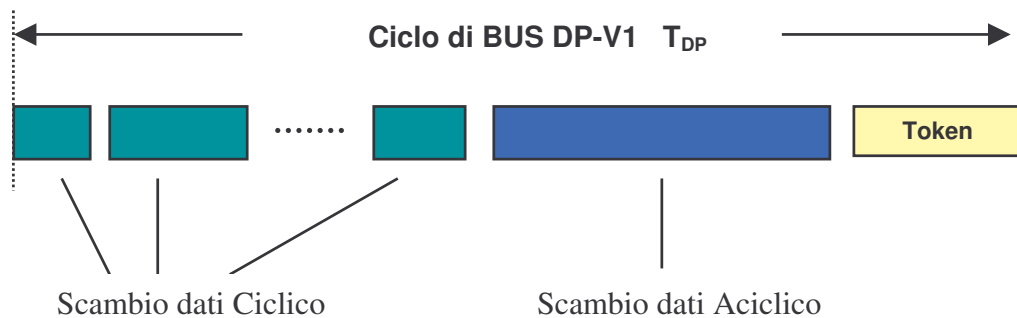



Figura B.6 – Ciclo di bus DP-V1

B.4.15 COMUNICAZIONE ISOCRONA (PROFIBUS DP-V2)

La comunicazione sincronizzata al ciclo di clock è implementata attraverso un segnale di clock isocrono. Questo ciclico, isocrono impulso di clock è trasmesso come un telegramma di global control dal master a tutte le stazioni del bus. Master e slave possono così sincronizzare le loro applicazioni con questo segnale.

Meccanismi speciali, presenti in ogni stazione, rendono stabile la comunicazione, anche nel caso di una breve mancanza del clock di sistema.

Per l'automazione, la comunicazione con ciclo di clock sincronizzato è la base per la sincronizzazione degli azionamenti. Con una base di tempi isocrona non viene solo implementato lo scambio di dati e di messaggi isocrono sul bus di sistema, ma anche gli algoritmi interni di controllo, come il controllo di corrente e di velocità di un azionamento, vengono sincronizzati al ciclo di clock.

B.4.16 COMUNICAZIONE SLAVE-TO-SLAVE (PROFIBUS DP-V2)

La comunicazione slave-to-slave è usata da PROFIBUS per lo scambio dati fra gli slave. E' una comunicazione basata sul modello Publisher-Subscriber. Il modello è basato su un publisher (slave) che fornisce i suoi dati non solo al master DP, ma anche alle altre stazioni (subscriber), in modo che tutti gli altri slave possono accedere ai suoi dati ed agire di conseguenza. Tutto questo deve essere definito attraverso la configurazione del sistema PROFIBUS. In fase di configurazione si definiscono le relazioni slave-to-slave tra gli slave DP, i dati che il publisher rende disponibili, e a quali dati il subscriber deve accedere. La comunicazione slave-to-slave è associata allo scambio dati ciclico di PROFIBUS DP.
La figura mostra lo scambio dati della comunicazione slave-to-slave.

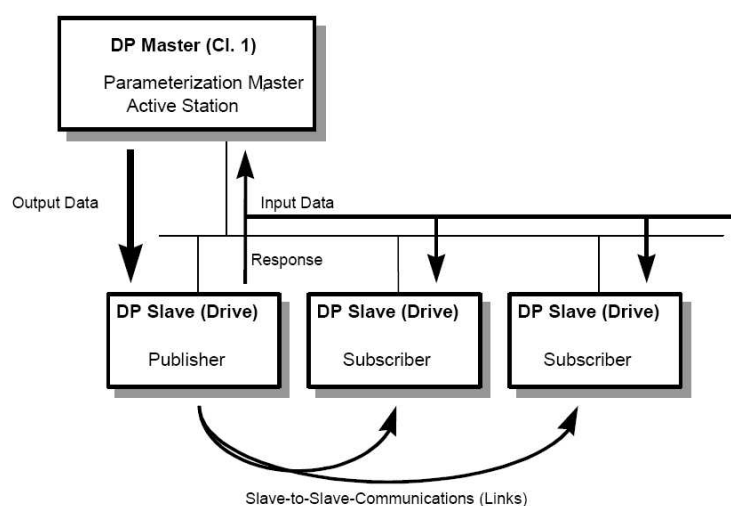



Figura B.7 – Comunicazione slave-to-slave

B.4.17 DEVICE DATABASE FILE (GSD)

Nel mondo sono stati sviluppati diversi dispositivi che seguono il protocollo PROFIBUS; essi differiscono sia per funzionalità che rendono disponibili (numero degli ingressi e delle uscite, tipologia dei messaggi diagnostici) che per parametri del bus che sono in grado di supportare. Per rendere semplice ed il più possibile facile la progettazione di un sistema PROFIBUS, a ogni dispositivo realizzato viene associata una scheda tecnica elettronica, nota con il nome di 'device database file' o 'GSD file'.

Essa contiene tutte le informazioni che un tool di progettazione richiede per implementare un sistema.

Sul sito web www.profibus.com sono disponibili i file GSD di tutti i dispositivi PROFIBUS certificati.

B.4.18 PROFILI PROFIBUS DP

Il protocollo PROFIBUS definisce come vengono scambiati i dati tra le stazioni presenti sul bus.

In aggiunta è possibile realizzare dei 'profili' che permettono di stabilire il significato dei dati scambiati sul bus soprattutto nelle parti definite dall'utente e relative a una specifica applicazione, e come il dispositivo PROFIBUS DP deve essere utilizzato in un certo ambito di applicazione.

L'utilizzo dei profili permette di ridurre il costo di ingegnerizzazione, visto che i parametri relativi all'applicazione sono specificati dettagliatamente.

Appendice C
PROFINET CBA

C.1 PROFINET CBA

Il protocollo PROFINET CBA (Component-based Automation) trova il suo campo applicativo nella realizzazione di impianti modulari.
Con PROFINET CBA, ogni parte dell'impianto, con il proprio sistema elettrico e meccanico, funziona come un modulo indipendente. Tutte le relazioni tra i singoli nodi modulari possono essere configurate graficamente tramite appositi tool senza la necessità di un laborioso lavoro di programmazione.
In questo modo, PROFINET CBA assicura massima flessibilità e produttività all'impianto per il suo intero ciclo di vita; permette di effettuare rapidi cambiamenti per quanto riguarda il piano di produzione e permette di ridurre il tempo di avviamento dell'impianto fornendo allo stesso tempo un completo supporto per la manutenzione e la gestione della qualità.

C.1.1 MODULI TECNOLOGICI

La funzione di un impianto automatico o di una macchina in un processo di manufacturing è realizzata tramite l'interazione tra meccanica, parte elettrico/elettronica e parte di controllo logico/software.
Lavorando a questo principio, PROFINET definisce le parti "meccaniche", "elettriche/elettronica" e "logica di controllo/software" in termini funzionali per formare un **modulo tecnologico** (Fig. C.1).

Figura C.1 - Modulo tecnologico in PROFINET

C.1.2 MODELLO DEL COMPONENTE DI PROFINET

I sistemi consistono solitamente di parecchie sotto unità, che si comportano come moduli tecnologici in gran parte autonomi e coordinano le loro interazioni per mezzo di un numero variabile di sincronizzazioni, di controlli di sequenza e di segnali di scambio di informazioni.

Il modello componente di PROFINET utilizza tali moduli tecnologici come base. PROFINET CBA usa la tecnologia componente come concetto base. L'intera funzionalità di un modulo tecnologico è incapsulata in un componente software (il componente di PROFINET). Il modello componente consente la combinazione delle applicazioni dalle origini differenti secondo il principio del blocchetto di costruzione e facilita notevolmente la programmazione di comunicazioni.

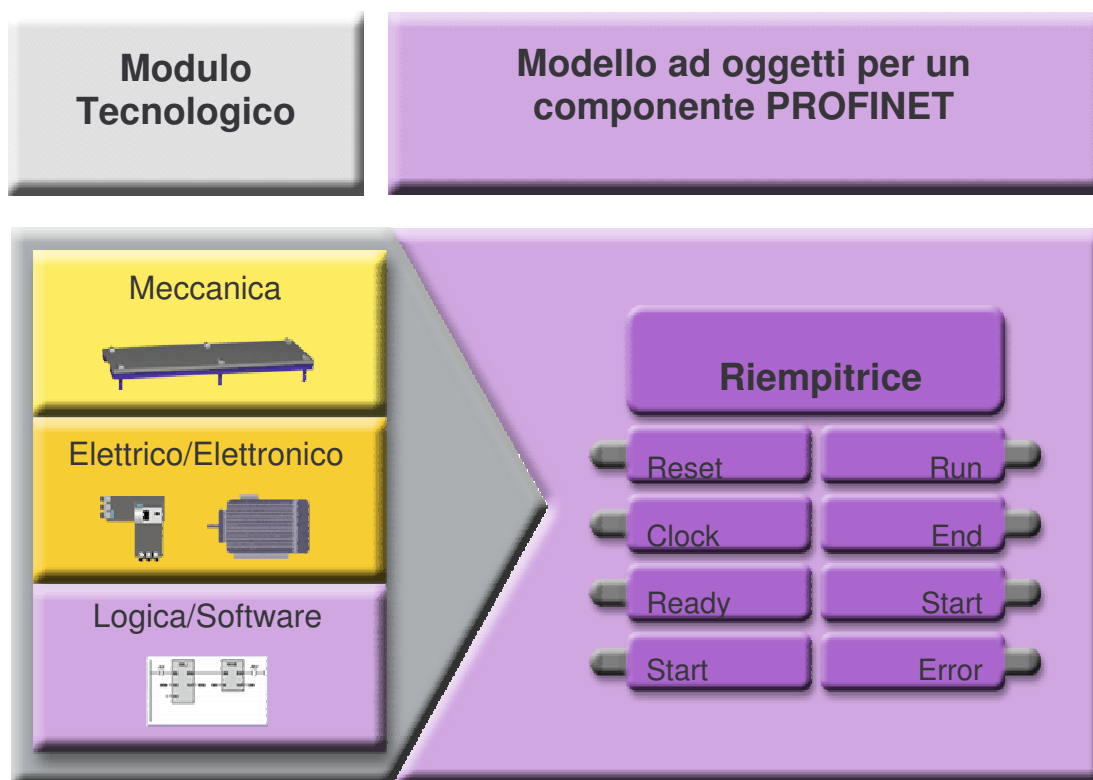



Figura C.2 - Modello componente in PROFINET

Un tal componente è modellato come un oggetto ed è da considerare come un scatola nera. Viene definita un' interfaccia componente per comunicare con altri componenti all'interno del sistema distribuito. Tramite l'interfaccia componente è possibile accedere soltanto a quelle variabili che sono necessarie per attuare interazione con altri componenti.

Nella fase di ingegneria di sistema i rapporti di comunicazione fra i componenti ed i loro dispositivi sono definiti configurando i collegamenti fra le interfacce componenti e un'applicazione specifica.

Progettare un sistema distribuito di automazione seguendo queste linee è un presupposto per la modularizzazione degli impianti e delle macchine.

C.1.2.1 VANTAGGI DELLA MODULARIZZAZIONE:

- Suddivisione della funzionalità globale in applicazioni parziali di maggiore trasparenza

- Maggiore disponibilità grazie a moduli funzionanti in modo autonomo
- Considerevole riduzione del tempo di messa in servizio grazie alla messa in servizio parallela dei moduli
- Software applicativo facilmente riutilizzabile
- Standardizzazione di macchine/impianti:
  Mediante i diversi moduli standard risulta possibile configurare in modo rapido la macchina/impianto specifico del Cliente

I costi di ingegneria di conseguenza sono notevolmente ridotti.

C.1.2.2 LIVELLO DI DETTAGLIO DEI MODULI TECNOLOGICI

Il livello di dettaglio usato nella definizione di un componente PROFINET può variare da un singolo dispositivo ad una macchina complessa composta da una molteplicità di dispositivi.
Dei moduli è importante considerare la loro riutilizzabilità in vari sistemi alla luce dei costi e della disponibilità. L'obiettivo è unire i diversi componenti per generare un sistema completo.
Va comunque considerato che: da una parte, un livello di dettaglio troppo alto produce una vista tecnologica dell'installazione che risulta più complessa e provoca un aumento dei costi di ingegneria. D'altra parte, un livello di dettaglio troppo basso riduce il grado di riutilizzabilità provocando un aumento dei costi di esecuzione.
Per ottenere il più alto grado possibile di riutilizzabilità per i diversi moduli è necessario esaminare dettagliatamente quale delle parti può essere usata diverse volte senza rendere necessario cambiare il relativo modulo standard.
I componenti software sono generati dal fornitore della macchina o dell'impianto.

C.1.3 LA DESCRIZIONE DEL COMPONENTE DI PROFINET (PCD)

I componenti PROFINET sono descritti in un file PCD (PROFINET Component Description) in formato XML.

Il file è creato utilizzando tool diversi a seconda del produttore dotati di un generatore di componenti. Alternativamente è possibile creare un file PCD usando il software PROFINET Component Editor, scaricabile dal sito PROFIBUS www.profibus.com.

Il file XML generato contiene le informazioni riguardo alla funzione ed agli oggetti dei componenti PROFINET.

In PROFINET il file componente XML contiene i seguenti dati**:**

- **descrizione dei componenti** come elementi di libreria:
  identificazione componente, nome componente
- **descrizione dell'hardware:**
  indirizzo IP, accesso ai dati diagnostici, download delle connessioni
- **descrizione della funzionalità del software:**
  assegnazione del software all'hardware, interfaccia componente, proprietà delle variabili come i loro nomi, tipi di dati, e direzione (input o output)
- **buffer dei componenti del progetto**
  le librerie dei componenti sono organizzate per permettere la riutilizzabilità dei componenti.

C.1.4 XML

L' Extensible Markup Language (XML) è una lingua di descrizione flessibile di dati. Questi contengono testo normale arricchito con informazioni di struttura.

Un documento di XML si compone degli **elementi** e degli **attributi**.

- Gli elementi possono contenere ulteriori elementi.
- Gli attributi sono assegnati agli elementi.
- La struttura si presenta con la scelta degli elementi e degli attributi.

Uno schema definisce così una lingua per la struttura dei dati in una area specifica di uso. La creazione dei documenti XML, degli schemi e della convalida e dell'elaborazione dei file è supportata da numerosi strumenti. Oggi, XML è supportato praticamente su tutti i sistemi.

C.1.4.1 REALIZZAZIONE DI UNA APPLICAZIONE PROFINET CBA COMPLETA

Un'applicazione viene generata in tre fasi:

1) **Creazione dei componenti**

I componenti sono generati come un'immagine dei moduli tecnologici presenti nell'impianto. La programmazione e la configurazione dei dispositivi sono effettuate, come prima, con i rispettivi tool di configurazione e di programmazione forniti dai singoli venditori. E' quindi possibile continuare ad utilizzare i programmi applicativi esistenti e l'esperienza dei programmatori e del personale addetto alla manutenzione.
Il software applicativo è incapsulato sotto forma di un componente di PROFINET, come una descrizione PCD (PROFINET Component Description) del componente. Una descrizione componente è generata così sotto forma di un file XML.
Queste descrizioni componenti sono importate nella libreria dell'editor delle connessioni.

2) **Connessione dei componenti attraverso un editor**

Usando l'editor delle connessioni di PROFINET i componenti PROFINET generati sono presi dalla libreria nella quale sono stati precedentemente salvati (drag and drop) e sono collegati fra loro con un semplice click del mouse a formare un'applicazione (figura C.3).

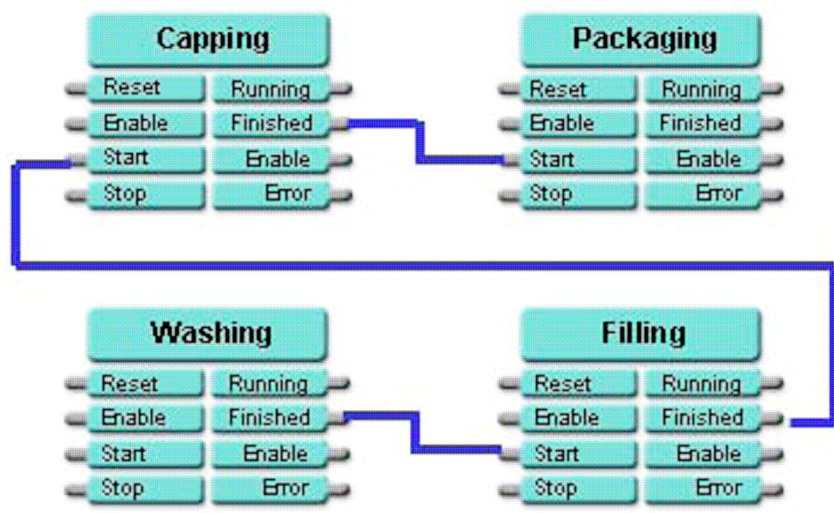



Figura C.3 – Collegamento dei vari moduli

L'editor delle connessioni visualizza l'insieme dei componenti PROFINET dei diversi moduli tecnologici che costituiscono l'applicazione distribuita.

L'editor funziona indipendentemente dal fornitore del componente, infatti può configurare componenti PROFINET di diversi fornitori.

L'interconnessione grafica del componente durante la configurazione, si tirano semplicemente dei "fili" fra i vari componenti PROFINET, porta ad una semplificazione del progetto di ingegnerizzazione e anche alcuni vantaggi importanti rispetto alla programmazione tradizionale di un sistema.

L'approccio usato fino ad oggi per definire il programma di comunicazione prevedeva infatti di:

- Conoscere:
  - quante variabili devono essere scambiate
  - chi sono i partners della comunicazione
  - quale rete di comunicazione utilizzare
- Configurare la comunicazione:

  La programmazione richiede la conoscenza dettagliata dell'integrazione e della sequenzialità delle funzioni di comunicazione nel dispositivo.

  Quando la programmazione è realizzata deve già essere chiaro quali dispositivi comunicheranno con ogni altro e a quali tempi ed attraverso quali sistemi del bus.

- definire il tipo di comunicazione (S7-Protocol, TCP/IP, ISO-Transport,...)

- configurare la comunicazione con il partner con Netpro

- Intervenire sul programma utente:

  - definire le aree di scambio dati per la Send-Receive

  - richiamare i blocchi di comunicazione e assegnare ID number e i parametri relativi

Queste operazioni non sono più necessarie qualora si lavori nello standard PROFINET. Nessuna conoscenza delle funzioni di comunicazione è richiesta per la configurazione. Quest'ultime sono effettuate automaticamente nei dispositivi.

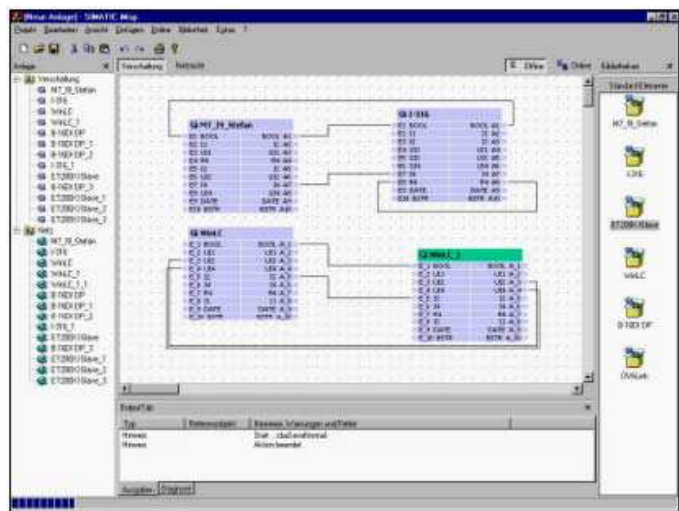



Figura C.5 - Vista del sistema nell'editor Imap di connessione

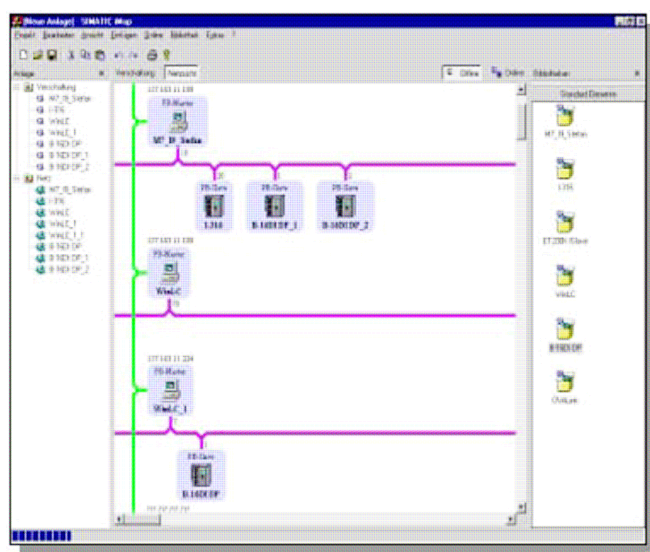



Figura C.6 - Vista della rete nell'editor Imap di connessione

3) Download

L'operazione di download assicura che la configurazione completa sia caricata nei dispositivi PROFINET.

Dopo aver realizzato le giuste connessioni tra i componenti , i dati del collegamento ed il codice così come i dati di configurazione dei componenti stessi vengono trasferiti nei dispositivi.

Ogni dispositivo conosce così tutti i relativi partners di comunicazione, i rapporti di comunicazione e le informazioni scambiabili.

L'operazione di download stessa avviene in tre punti:

- Un controllo preventivo per assicurarsi che tutti i dati da caricare siano consistenti.

  In particolare sono controllati i collegamenti configurati fra i partner al fine di stabilire se essi possano realmente essere effettuati.
- Il download dei dati di configurazione e il codice.
- Il download dei collegamenti.

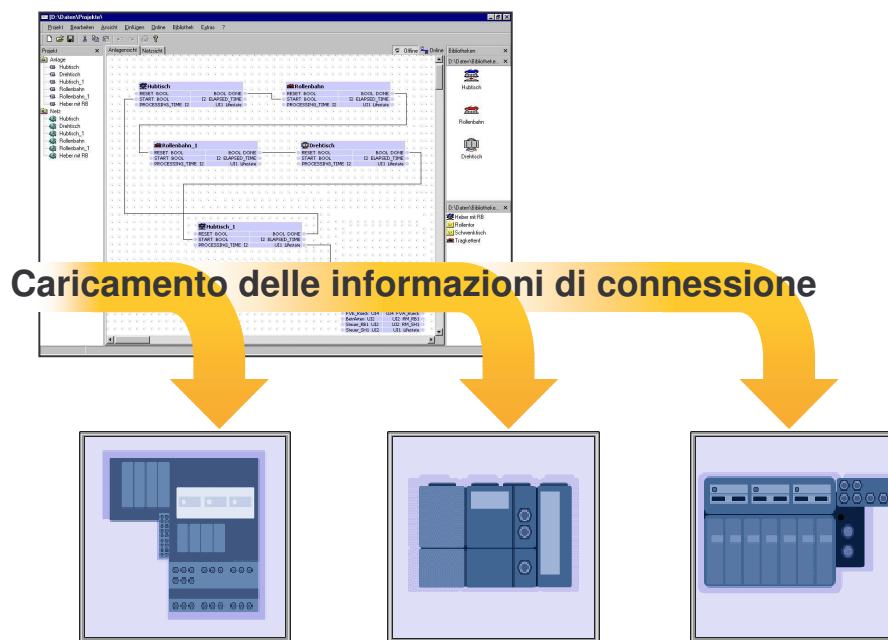



Figura C.7 Downloading dei dati nei dispositivi PROFINET

A questo punto l'applicazione distribuita può allora essere fatta funzionare.

BIBLIOGRAFIA

[1] Manfred Pop, (2003):
The New Rapid Way to PROFIBUS DP, PROFIBUS Nutzorganisation

[2] Manfred Pop, Karl Weber, (2004):
The Rapid Way to PROFINET, PROFIBUS Nutzorganisation

[3] Coutinho j., Martin S., Samata G., Tapley S., Wilkin D.,(1995):
Fieldbus Tutorial

[4] Mandada M.:
I Bus di Campo : dalla Teoria alla Pratica

[5] Andrew S.Tanembaum, (1991):
Reti di computer,2° edizione, Hall & Jackson

[6] *PROFIBUS Technical Overview*, www.PROFIBUS.com

[7] *PROFINET Technical Overview*, www.PROFIBUS.com

[8] PROFIBUS day 26 giugno 2003:Antonio Augelli SIEMENS s.p.
www.PROFIBUS.com

[9] Flammini A :
Dispensa *Sistemi per l'automazione e PLC*
yoda.ing.unibs.it/Materiale_didattico/Sistemi_per_l'automazione_e_PLC

[10] Zantedeschi Corrado

Tesi di laurea: Progettazione e realizzazione a microprocessore di un interfaccia PROFIBUS per applicazioni industriali anno accademico 97/98

[11] Zanetti Damiano

Tesi di laurea: Realizzazione di una rete in Profibus DP anno accademico 2002/2003

[12] Barbo' Sauro

Tesi di laurea: Realizzazione e caratterizzazione di una rete PROFINET anno accademico 2002/2003

[13] P.Ferrari, A.Flammini, D. Marioli, A. Taroni

Experimental evaluation of PROFINET performance. WFCS 2004 Proceedings. Ed. Thilo Sauter and Francisco Vasques

[14] J. Feld

PROFINET – Scalable Factory Communication for all Application. WFCS 2004 Proceedings. Ed. Thilo Sauter and Francisco Vasques

[15] F. Bertozzi, M. Di Natale, L.Almeida

Amission control and overload handling in FTT-CAN. WFCS 2004 Proceedings. Ed. Thilo Sauter and Francisco Vasques

[16] M. Calha, J. Fonseca

Approaches to the FTT-based Sheduling of task and message. WFCS 2004 Proceedings. Ed. Thilo Sauter and Francisco Vasques

[17] G.Rodriguez-Navas, J. Rigo, J. Proenza, J.Ferreira, L.Almeida, J. Fonseca

Design and Modelling of a protocol to enforce consistency among replicated masters in FTT-CAN. WFCS 2004 Proceedings. Ed. Thilo Sauter and Francisco Vasques

[18] PROFIBUS INTERNATIONAL

*PROFIdrive-Profil Drive Tecnology,*April 2004, PROFIBUS INTERNATIONAL

[19] Siemens SIMATIC

Sistema di periferia decentrata ET 200S

[20] *Periferia decentrata ET 200S Manuale tecnico* www.siemens.com

[21] Siemens SIMATIC

ET 200S Moduli d'interfaccia IM 151/cpu

[22] Siemens SIMATIC

Sistema di automazione S7-300 Configurazione e installazione

[23] Siemens SIMATIC

Software di sistema per S7-300/400 Funzioni standard e di sistema Manuale di riferimento

[24] Siemens SIMATIC

Software di sistema per S7-300/400 Funzioni standard parte 2

[25] Siemens SIMATIC

Sistema di automazione S7-300 Caratteristiche delle unità modulari

[26] Siemens SIMATIC

Sistema di automazione S7-300 Installazione, configurazione e dati della CPU

[27] Siemens SIMATIC

Programmazione con STEP 7 V5.3

[28] Siemens SIMATIC
S7-300 Manuale tecnico

[29] Siemens SIMATIC NET
NCM S7 per PROFIBUS Manuale Volume 1 di 2

[30] Siemens SIMATIC NET
Reti PROFINET Manuale

[31] Siemens SIMATIC
Configurazione dell'hardware e progettazione collegamenti STEP 7 V53. Manuale

[32] Siemens SIMATIC NET
DP Base Programming Interface for CP5613/5614

[33] Siemens SIMATIC NET
Product Information/Installation Instructions CP5613

[34] Siemens SIMATIC
Component based Automation Commissioning Systems Tutorial

[35] Siemens SIMATIC
Commissioning PC Station- Manual/Quick Start

[36] www.siemens.com